```
[notebook]
# ! [ ] [ ] [ ]
```

```
# Model 2: DenseNet
import os
import numpy as np
import pandas as pd
from sklearn.model_selection import StratifiedKfold
from keras.preprocessing.image import ImageDataGenerator
from keras.models import *
from keras.layers import *
from keras.callbacks import EarlyStopping, ReduceLROnPlateau, CsvLogger
from keras.applications.densenet import DenseNet121, preprocess_input

# Kaggle Kernels Relevant Paths
image_input = '/kaggle/input/iwildcam-2019-fgvc6/train/'

# Local Laptop Paths
base_path = '/home/ubuntu/Desktop/kaggle/iwildcam/'
# base_path = '/home/ubuntu/Desktop/kaggle/iwildcam-2019-fgvc6/'

# Image Input
image_input = base_path+'iwildcam-2019-fgvc6/train/'
# image_input = base_path+'train/'

# Model Output Paths
model_output = base_path+'models/'

# Submission Output Paths
submission_output = base_path+'submissions/'

# Log Output Paths
log_output = base_path+'logs/'

# Checkpoint Output Paths
checkpoint_output = base_path+'checkpoints/'

# Tensorboard Output Paths
tensorboard_output = base_path+'tensorboard/'

# Load Training Data
df = pd.read_csv(base_path+'iwildcam-2019-fgvc6/train.csv')

# Load Validation Data
df_val = pd.read_csv(base_path+'iwildcam-2019-fgvc6/validation.csv')

# Count Number of Training Classes
num_classes = len(df['category_id'].unique)

# Print Shape of Training and Validation Data
print('Training Data Shape: {}'.format(df.shape))
print('Validation Data Shape: {}'.format(df_val.shape))

# Save Image Paths for Training and Validation
df['image_path'] = df.apply(lambda x: (image_input + str(x['id']) + '.jpg'), axis=1)
df_val['image_path'] = df_val.apply(lambda x: (image_input + str(x['id']) + '.jpg'), axis=1)

# Print Training and Validation Data with Image Paths
print('Training Data With Image Path Shape: {}'.format(df.shape))
print('Validation Data With Image Path Shape: {}'.format(df_val.shape))

# Print Head of Training and Validation Data
print('Training Data With Image Path Head: ', df.head())
print('Validation Data With Image Path Head: ', df_val.head())

# Hyperparameters
batch_size = 256
epochs = 30
learning_rate = 0.0001
decay_rate = learning_rate / epochs
momentum = 0.8
verbose = 1
early_stopping_patience = 2
reduce_lr_plateau_factor = 0.1
reduce_lr_plateau_patience = 1
reduce_lr_plateau_min_lr = 0.00001
reduce_lr_plateau_epsilon = 0.0001
csv_log_filename = 'densenet_v1.log'
checkpoint_filepath = 'densenet_v1_{epoch:02d}-{val_loss:.4f}.hdf5'
tensorboard_log_dir = '/home/ubuntu/Desktop/kaggle/iwildcam/tensorboard/densenet_v1'

# Image Generator
train_datagen = ImageDataGenerator(preprocessing_function=preprocess_input,
                                   horizontal_flip=True,
                                   vertical_flip=True,
                                   rotation_range=180,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   zoom_range=0.2)

train_generator = train_datagen.flow_from_dataframe(df,
                                    directory=None,
                                    x_col='image_path',
                                    y_col='category_id',
                                    class_mode='raw',
                                    batch_size=batch_size,
                                    shuffle=True)

val_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)

val_generator = val_datagen.flow_from_dataframe(df_val,
                                    directory=None,
                                    x_col='image_path',
                                    y_col='category_id',
                                    class_mode='raw',
                                    batch_size=batch_size,
                                    shuffle=False)

# Callbacks
early_stopping = EarlyStopping(patience=early_stopping_patience, verbose=verbose)

checkpoint = ModelCheckpoint(checkpoint_output+checkpoint_filepath,
                             monitor='val_loss',
                             verbose=verbose,
                             save_best_only=False,
                             save_weights_only=False,
                             mode='auto',
                             period=1)

csv_logger = CsvLogger(log_output+csv_log_filename, append=True, separator=',')

reduce_lr = ReduceLROnPlateau(monitor='val_loss',
                              factor=reduce_lr_plateau_factor,
                              patience=reduce_lr_plateau_patience,
                              verbose=verbose,
                              mode='auto',
                              epsilon=reduce_lr_plateau_epsilon,
                              cooldown=0,
                              min_lr=reduce_lr_plateau_min_lr)

tensorboard = TensorBoard(log_dir=tensorboard_log_dir,
                          histogram_freq=0,
                          batch_size=batch_size,
                          write_graph=True,
                          write_grad=True,
                          write_images=True,
                          embeddings_freq=0,
                          embeddings_layer_names=None,
                          embeddings_metadata=None,
                          embeddings_data=None,
                          update_freq='epoch')

# Initialize Model
model = DenseNet121(weights=None,
                    input_shape=(224,224,3),
                    pooling=None,
                    classes=num_classes)

# Compile Model
model.compile(loss='mean_squared_error',
              optimizer=Adam(lr=learning_rate, beta_1=0.9, beta_2=0.999, epsilon=None, decay=decay_rate, amsgrad=False),
              metrics=['accuracy'])

model.summary()

# Fit Model
model.fit_generator(train_generator,
                    steps_per_epoch=len(df)//batch_size,
                    epochs=epochs,
                    verbose=verbose,
                    callbacks=[early_stopping, checkpoint, csv_logger, reduce_lr, tensorboard],
                    validation_data=val_generator,
                    validation_steps=len(df_val)//batch_size,
                    class_weight=None,
                    max_queue_size=10,
                    workers=1,
                    use_multiprocessing=False,
                    shuffle=True,
                    initial_epoch=0)

# Fine-Tuning
for layer in model.layers:
    layer.trainable = True
    if hasattr(layer, 'moving_mean'):
        layer.moving_mean = False
    if hasattr(layer, 'moving_variance'):
        layer.moving_variance = False

# Compile Model
model.compile(loss='mean_squared_error',
              optimizer=Adam(lr=learning_rate, beta_1=0.9, beta_2=0.999, epsilon=None, decay=decay_rate, amsgrad=False),
              metrics=['accuracy'])

model.summary()

# Fit Model
model.fit_generator(train_generator,
                    steps_per_epoch=len(df)//batch_size,
                    epochs=epochs,
                    verbose=verbose,
                    callbacks=[early_stopping, checkpoint, csv_logger, reduce_lr, tensorboard],
                    validation_data=val_generator,
                    validation_steps=len(df_val)//batch_size,
                    class_weight=None,
                    max_queue_size=10,
                    workers=1,
                    use_multiprocessing=False,
                    shuffle=True,
                    initial_epoch=0)
```

```
Model: "densenet_2"
```

Layer (type)	Output Shape	Param #	Connected to
input_1(InputLayer)	(None, 224, 224, 3)	0	
zero_pad(ZeroPadding2D)	(None, 230, 230, 3)	0	input_1[0][0]
conv1/conv(Conv2D)	(None, 112, 112, 64)	9,408	zero_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]
zero_pad(ZeroPadding2D)	(None, 114, 114, 64)	0	conv1/relu[0][0]
pool1(MaxPooling2D)	(None, 56, 56, 64)	0	zero_pad[0][0]
conv2_block1_0_bn(BatchNormalization)	(None, 56, 56, 64)	256	pool1[0][0]
conv2_block1_0_relu(Activation)	(None, 56, 56, 64)	0	conv2_block1_0_bn[0][0]
conv2_block1_1_conv(Conv2D)	(None, 56, 56, 128)	8,192	conv2_block1_0_relu[0][0]
conv2_block1_1_bn(BatchNormalization)	(None, 56, 56, 128)	512	conv2_block1_1_conv[0][0]
conv2_block1_1_relu(Activation)	(None, 56, 56, 128)	0	conv2_block1_1_bn[0][0]
conv2_block1_2_conv(Conv2D)	(None, 56, 56, 32)	36,864	conv2_block1_1_relu[0][0]
conv2_block1_concat(Concatenate)	(None, 56, 56, 64)	0	pool1[0][0] conv2_block1_2_conv[0][0]
conv2_block2_0_bn(BatchNormalization)	(None, 56, 56, 64)	256	conv2_block1_concat[0][0]
conv2_block2_0_relu(Activation)	(None, 56, 56, 64)	0	conv2_block2_0_bn[0][0]
conv2_block2_1_conv(Conv2D)	(None, 56, 56, 128)	8,192	conv2_block2_0_relu[0][0]
conv2_block2_1_bn(BatchNormalization)	(None, 56, 56, 128)	512	conv2_block2_1_conv[0][0]
conv2_block2_1_relu(Activation)	(None, 56, 56, 128)	0	conv2_block2_1_bn[0][0]
conv2_block2_2_conv(Conv2D)	(None, 56, 56, 32)	36,864	conv2_block2_1_relu[0][0]
conv2_block2_concat(Concatenate)	(None, 56, 56, 96)	0	conv2_block1_concat[0][0] conv2_block2_2_conv[0][0]
conv2_block3_0_bn(BatchNormalization)	(None, 56, 56, 96)	384	conv2_block2_concat[0][0]
conv2_block3_0_relu(Activation)	(None, 56, 56, 96)	0	conv2_block3_0_bn[0][0]
conv2_block3_1_conv(Conv2D)	(None, 56, 56, 128)	12,288	conv2_block3_0_relu[0][0]
conv2_block3_1_bn(BatchNormalization)	(None, 56, 56, 128)	512	conv2_block3_1_conv[0][0]
conv2_block3_1_relu(Activation)	(None, 56, 56, 128)	0	conv2_block3_1_bn[0][0]
conv2_block3_2_conv(Conv2D)	(None, 56, 56, 32)	36,864	conv2_block3_1_relu[0][0]
conv2_block3_concat(Concatenate)	(None, 56, 56, 128)	0	conv2_block2_concat[0][0] conv2_block3_2_conv[0][0]
conv2_block4_0_bn(BatchNormalization)	(None, 56, 56, 128)	512	conv2_block3_concat[0][0]
conv2_block4_0_relu(Activation)	(None, 56, 56, 128)	0	conv2_block4_0_bn[0][0]
conv2_block4_1_conv(Conv2D)	(None, 56, 56, 128)	16,384	conv2_block4_0_relu[0][0]
conv2_block4_1_bn(BatchNormalization)	(None, 56, 56, 128)	512	conv2_block4_1_conv[0][0]
conv2_block4_1_relu(Activation)	(None, 56, 56, 128)	0	conv2_block4_1_bn[0][0]
conv2_block4_2_conv(Conv2D)	(None, 56, 56, 32)	36,864	conv2_block4_1_relu[0][0]
conv2_block4_concat(Concatenate)	(None, 56, 56, 160)	0	conv2_block3_concat[0][0] conv2_block4_2_conv[0][0]
conv2_block5_0_bn(BatchNormalization)	(None, 56, 56, 160)	640	conv2_block4_concat[0][0]
conv2_block5_0_relu(Activation)	(None, 56, 56, 160)	0	conv2_block5_0_bn[0][0]
conv2_block5_1_conv(Conv2D)	(None, 56, 56, 128)	20,480	conv2_block5_0_relu[0][0]
conv2_block5_1_bn(BatchNormalization)	(None, 56, 56, 128)	512	conv2_block5_1_conv[0][0]
conv2_block5_1_relu(Activation)	(None, 56, 56, 128)	0	conv2_block5_1_bn[0][0]
conv2_block5_2_conv(Conv2D)	(None, 56, 56, 32)	36,864	conv2_block5_1_relu[0][0]
conv2_block5_concat(Concatenate)	(None, 56, 56, 192)	0	conv2_block4_concat[0][0] conv2_block5_2_conv[0][0]
conv2_block6_0_bn(BatchNormalization)	(None, 56, 56, 192)	768	conv2_block5_concat[0][0]
conv2_block6_0_relu(Activation)	(None, 56, 56, 192)	0	conv2_block6_0_bn[0][0]
conv2_block6_1_conv(Conv2D)	(None, 56, 56, 128)	24,576	conv2_block6_0_relu[0][0]
conv2_block6_1_bn(BatchNormalization)	(None, 56, 56, 128)	512	conv2_block6_1_conv[0][0]
conv2_block6_1_relu(Activation)	(None, 56, 56, 128)	0	conv2_block6_1_bn[0][0]
conv2_block6_2_conv(Conv2D)	(None, 56, 56, 32)	36,864	conv2_block6_1_relu[0][0]
conv2_block6_concat(Concatenate)	(None, 56, 56, 224)	0	conv2_block5_concat[0][0] conv2_block6_2_conv[0][0]
pool2_bn(BatchNormalization)	(None, 56, 56, 224)	896	conv2_block6_concat[0][0]
pool2_relu(Activation)	(None, 56, 56, 224)	0	pool2_bn[0][0]
pool2_conv(Conv2D)	(None, 56, 56, 112)	25,088	pool2_relu[0][0]
pool2_drop(Dropout)	(None, 56, 56, 112)	0	pool2_conv[0][0]
pool2_pool(AveragePooling2D)	(None, 28, 28, 112)	0	pool2_drop[0][0]
conv3_block1_0_bn(BatchNormalization)	(None, 28, 28, 112)	448	pool2_pool[0][0]
conv3_block1_0_relu(Activation)	(None, 28, 28, 112)	0	conv3_block1_0_bn[0][0]
conv3_block1_1_conv(Conv2D)	(None, 28, 28, 128)	14,336	conv3_block1_0_relu[0][0]
conv3_block1_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block1_1_conv[0][0]
conv3_block1_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block1_1_bn[0][0]
conv3_block1_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block1_1_relu[0][0]
conv3_block1_concat(Concatenate)	(None, 28, 28, 144)	0	pool2_pool[0][0] conv3_block1_2_conv[0][0]
conv3_block2_0_bn(BatchNormalization)	(None, 28, 28, 144)	576	conv3_block1_concat[0][0]
conv3_block2_0_relu(Activation)	(None, 28, 28, 144)	0	conv3_block2_0_bn[0][0]
conv3_block2_1_conv(Conv2D)	(None, 28, 28, 128)	18,432	conv3_block2_0_relu[0][0]
conv3_block2_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block2_1_conv[0][0]
conv3_block2_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block2_1_bn[0][0]
conv3_block2_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block2_1_relu[0][0]
conv3_block2_concat(Concatenate)	(None, 28, 28, 176)	0	conv3_block1_concat[0][0] conv3_block2_2_conv[0][0]
conv3_block3_0_bn(BatchNormalization)	(None, 28, 28, 176)	704	conv3_block2_concat[0][0]
conv3_block3_0_relu(Activation)	(None, 28, 28, 176)	0	conv3_block3_0_bn[0][0]
conv3_block3_1_conv(Conv2D)	(None, 28, 28, 128)	22,528	conv3_block3_0_relu[0][0]
conv3_block3_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block3_1_conv[0][0]
conv3_block3_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block3_1_bn[0][0]
conv3_block3_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block3_1_relu[0][0]
conv3_block3_concat(Concatenate)	(None, 28, 28, 208)	0	conv3_block2_concat[0][0] conv3_block3_2_conv[0][0]
conv3_block4_0_bn(BatchNormalization)	(None, 28, 28, 208)	832	conv3_block3_concat[0][0]
conv3_block4_0_relu(Activation)	(None, 28, 28, 208)	0	conv3_block4_0_bn[0][0]
conv3_block4_1_conv(Conv2D)	(None, 28, 28, 128)	26,624	conv3_block4_0_relu[0][0]
conv3_block4_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block4_1_conv[0][0]
conv3_block4_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block4_1_bn[0][0]
conv3_block4_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block4_1_relu[0][0]
conv3_block4_concat(Concatenate)	(None, 28, 28, 240)	0	conv3_block3_concat[0][0] conv3_block4_2_conv[0][0]
conv3_block5_0_bn(BatchNormalization)	(None, 28, 28, 240)	960	conv3_block4_concat[0][0]
conv3_block5_0_relu(Activation)	(None, 28, 28, 240)	0	conv3_block5_0_bn[0][0]
conv3_block5_1_conv(Conv2D)	(None, 28, 28, 128)	30,720	conv3_block5_0_relu[0][0]
conv3_block5_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block5_1_conv[0][0]
conv3_block5_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block5_1_bn[0][0]
conv3_block5_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block5_1_relu[0][0]
conv3_block5_concat(Concatenate)	(None, 28, 28, 272)	0	conv3_block4_concat[0][0] conv3_block5_2_conv[0][0]
conv3_block6_0_bn(BatchNormalization)	(None, 28, 28, 272)	1,088	conv3_block5_concat[0][0]
conv3_block6_0_relu(Activation)	(None, 28, 28, 272)	0	conv3_block6_0_bn[0][0]
conv3_block6_1_conv(Conv2D)	(None, 28, 28, 128)	34,816	conv3_block6_0_relu[0][0]
conv3_block6_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block6_1_conv[0][0]
conv3_block6_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block6_1_bn[0][0]
conv3_block6_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block6_1_relu[0][0]
conv3_block6_concat(Concatenate)	(None, 28, 28, 304)	0	conv3_block5_concat[0][0] conv3_block6_2_conv[0][0]
conv3_block7_0_bn(BatchNormalization)	(None, 28, 28, 304)	1,216	conv3_block6_concat[0][0]
conv3_block7_0_relu(Activation)	(None, 28, 28, 304)	0	conv3_block7_0_bn[0][0]
conv3_block7_1_conv(Conv2D)	(None, 28, 28, 128)	38,912	conv3_block7_0_relu[0][0]
conv3_block7_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block7_1_conv[0][0]
conv3_block7_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block7_1_bn[0][0]
conv3_block7_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block7_1_relu[0][0]
conv3_block7_concat(Concatenate)	(None, 28, 28, 336)	0	conv3_block6_concat[0][0] conv3_block7_2_conv[0][0]
conv3_block8_0_bn(BatchNormalization)	(None, 28, 28, 336)	1,344	conv3_block7_concat[0][0]
conv3_block8_0_relu(Activation)	(None, 28, 28, 336)	0	conv3_block8_0_bn[0][0]
conv3_block8_1_conv(Conv2D)	(None, 28, 28, 128)	43,008	conv3_block8_0_relu[0][0]
conv3_block8_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block8_1_conv[0][0]
conv3_block8_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block8_1_bn[0][0]
conv3_block8_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block8_1_relu[0][0]
conv3_block8_concat(Concatenate)	(None, 28, 28, 368)	0	conv3_block7_concat[0][0] conv3_block8_2_conv[0][0]
conv3_block9_0_bn(BatchNormalization)	(None, 28, 28, 368)	1,472	conv3_block8_concat[0][0]
conv3_block9_0_relu(Activation)	(None, 28, 28, 368)	0	conv3_block9_0_bn[0][0]
conv3_block9_1_conv(Conv2D)	(None, 28, 28, 128)	47,104	conv3_block9_0_relu[0][0]
conv3_block9_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block9_1_conv[0][0]
conv3_block9_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block9_1_bn[0][0]
conv3_block9_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block9_1_relu[0][0]
conv3_block9_concat(Concatenate)	(None, 28, 28, 400)	0	conv3_block8_concat[0][0] conv3_block9_2_conv[0][0]
conv3_block10_0_bn(BatchNormalization)	(None, 28, 28, 400)	1,600	conv3_block9_concat[0][0]
conv3_block10_0_relu(Activation)	(None, 28, 28, 400)	0	conv3_block10_0_bn[0][0]
conv3_block10_1_conv(Conv2D)	(None, 28, 28, 128)	51,200	conv3_block10_0_relu[0][0]
conv3_block10_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block10_1_conv[0][0]
conv3_block10_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block10_1_bn[0][0]
conv3_block10_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block10_1_relu[0][0]
conv3_block10_concat(Concatenate)	(None, 28, 28, 432)	0	conv3_block9_concat[0][0] conv3_block10_2_conv[0][0]
conv3_block11_0_bn(BatchNormalization)	(None, 28, 28, 432)	1,728	conv3_block10_concat[0][0]
conv3_block11_0_relu(Activation)	(None, 28, 28, 432)	0	conv3_block11_0_bn[0][0]
conv3_block11_1_conv(Conv2D)	(None, 28, 28, 128)	55,296	conv3_block11_0_relu[0][0]
conv3_block11_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block11_1_conv[0][0]
conv3_block11_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block11_1_bn[0][0]
conv3_block11_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block11_1_relu[0][0]
conv3_block11_concat(Concatenate)	(None, 28, 28, 464)	0	conv3_block10_concat[0][0] conv3_block11_2_conv[0][0]
conv3_block12_0_bn(BatchNormalization)	(None, 28, 28, 464)	1,856	conv3_block11_concat[0][0]
conv3_block12_0_relu(Activation)	(None, 28, 28, 464)	0	conv3_block12_0_bn[0][0]
conv3_block12_1_conv(Conv2D)	(None, 28, 28, 128)	59,392	conv3_block12_0_relu[0][0]
conv3_block12_1_bn(BatchNormalization)	(None, 28, 28, 128)	512	conv3_block12_1_conv[0][0]
conv3_block12_1_relu(Activation)	(None, 28, 28, 128)	0	conv3_block12_1_bn[0][0]
conv3_block12_2_conv(Conv2D)	(None, 28, 28, 32)	36,864	conv3_block12_1_relu[0][0]
conv3_block12_concat(Concatenate)	(None, 28, 28, 496)	0	conv3_block11_concat[0][0] conv3_block12_2_conv[0][0]
pool3_bn(BatchNormalization)	(None, 28, 28, 496)	1,984	conv3_block12_concat[0][0]
pool3_relu(Activation)	(None, 28, 28, 496)	0	pool3_bn[0][0]
pool3_conv(Conv2D)	(None, 28, 28, 248)	123,008	pool3_relu[0][0]
pool3_drop(Dropout)	(None, 28, 28, 248)	0	pool3_conv[0][0]
pool3_pool(AveragePooling2D)	(None, 14, 14, 248)	0	pool3_drop[0][0]
conv4_block1_0_bn(BatchNormalization)	(None, 14, 14, 248)	992	pool3_pool[0][0]
conv4_block1_0_relu(Activation)	(None, 14, 14, 248)	0	conv4_block1_0_bn[0][0]
conv4_block1_1_conv(Conv2D)	(None, 14, 14, 128)	31,744	conv4_block1_0_relu[0][0]
conv4_block1_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block1_1_conv[0][0]
conv4_block1_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block1_1_bn[0][0]
conv4_block1_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block1_1_relu[0][0]
conv4_block1_concat(Concatenate)	(None, 14, 14, 280)	0	pool3_pool[0][0] conv4_block1_2_conv[0][0]
conv4_block2_0_bn(BatchNormalization)	(None, 14, 14, 280)	1,120	conv4_block1_concat[0][0]
conv4_block2_0_relu(Activation)	(None, 14, 14, 280)	0	conv4_block2_0_bn[0][0]
conv4_block2_1_conv(Conv2D)	(None, 14, 14, 128)	35,840	conv4_block2_0_relu[0][0]
conv4_block2_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block2_1_conv[0][0]
conv4_block2_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block2_1_bn[0][0]
conv4_block2_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block2_1_relu[0][0]
conv4_block2_concat(Concatenate)	(None, 14, 14, 312)	0	conv4_block1_concat[0][0] conv4_block2_2_conv[0][0]
conv4_block3_0_bn(BatchNormalization)	(None, 14, 14, 312)	1,248	conv4_block2_concat[0][0]
conv4_block3_0_relu(Activation)	(None, 14, 14, 312)	0	conv4_block3_0_bn[0][0]
conv4_block3_1_conv(Conv2D)	(None, 14, 14, 128)	39,936	conv4_block3_0_relu[0][0]
conv4_block3_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block3_1_conv[0][0]
conv4_block3_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block3_1_bn[0][0]
conv4_block3_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block3_1_relu[0][0]
conv4_block3_concat(Concatenate)	(None, 14, 14, 344)	0	conv4_block2_concat[0][0] conv4_block3_2_conv[0][0]
conv4_block4_0_bn(BatchNormalization)	(None, 14, 14, 344)	1,376	conv4_block3_concat[0][0]
conv4_block4_0_relu(Activation)	(None, 14, 14, 344)	0	conv4_block4_0_bn[0][0]
conv4_block4_1_conv(Conv2D)	(None, 14, 14, 128)	44,032	conv4_block4_0_relu[0][0]
conv4_block4_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block4_1_conv[0][0]
conv4_block4_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block4_1_bn[0][0]
conv4_block4_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block4_1_relu[0][0]
conv4_block4_concat(Concatenate)	(None, 14, 14, 376)	0	conv4_block3_concat[0][0] conv4_block4_2_conv[0][0]
conv4_block5_0_bn(BatchNormalization)	(None, 14, 14, 376)	1,504	conv4_block4_concat[0][0]
conv4_block5_0_relu(Activation)	(None, 14, 14, 376)	0	conv4_block5_0_bn[0][0]
conv4_block5_1_conv(Conv2D)	(None, 14, 14, 128)	48,128	conv4_block5_0_relu[0][0]
conv4_block5_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block5_1_conv[0][0]
conv4_block5_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block5_1_bn[0][0]
conv4_block5_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block5_1_relu[0][0]
conv4_block5_concat(Concatenate)	(None, 14, 14, 408)	0	conv4_block4_concat[0][0] conv4_block5_2_conv[0][0]
conv4_block6_0_bn(BatchNormalization)	(None, 14, 14, 408)	1,632	conv4_block5_concat[0][0]
conv4_block6_0_relu(Activation)	(None, 14, 14, 408)	0	conv4_block6_0_bn[0][0]
conv4_block6_1_conv(Conv2D)	(None, 14, 14, 128)	52,224	conv4_block6_0_relu[0][0]
conv4_block6_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block6_1_conv[0][0]
conv4_block6_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block6_1_bn[0][0]
conv4_block6_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block6_1_relu[0][0]
conv4_block6_concat(Concatenate)	(None, 14, 14, 440)	0	conv4_block5_concat[0][0] conv4_block6_2_conv[0][0]
conv4_block7_0_bn(BatchNormalization)	(None, 14, 14, 440)	1,760	conv4_block6_concat[0][0]
conv4_block7_0_relu(Activation)	(None, 14, 14, 440)	0	conv4_block7_0_bn[0][0]
conv4_block7_1_conv(Conv2D)	(None, 14, 14, 128)	56,320	conv4_block7_0_relu[0][0]
conv4_block7_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block7_1_conv[0][0]
conv4_block7_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block7_1_bn[0][0]
conv4_block7_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block7_1_relu[0][0]
conv4_block7_concat(Concatenate)	(None, 14, 14, 472)	0	conv4_block6_concat[0][0] conv4_block7_2_conv[0][0]
conv4_block8_0_bn(BatchNormalization)	(None, 14, 14, 472)	1,888	conv4_block7_concat[0][0]
conv4_block8_0_relu(Activation)	(None, 14, 14, 472)	0	conv4_block8_0_bn[0][0]
conv4_block8_1_conv(Conv2D)	(None, 14, 14, 128)	60,416	conv4_block8_0_relu[0][0]
conv4_block8_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block8_1_conv[0][0]
conv4_block8_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block8_1_bn[0][0]
conv4_block8_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block8_1_relu[0][0]
conv4_block8_concat(Concatenate)	(None, 14, 14, 504)	0	conv4_block7_concat[0][0] conv4_block8_2_conv[0][0]
conv4_block9_0_bn(BatchNormalization)	(None, 14, 14, 504)	2,016	conv4_block8_concat[0][0]
conv4_block9_0_relu(Activation)	(None, 14, 14, 504)	0	conv4_block9_0_bn[0][0]
conv4_block9_1_conv(Conv2D)	(None, 14, 14, 128)	64,512	conv4_block9_0_relu[0][0]
conv4_block9_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block9_1_conv[0][0]
conv4_block9_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block9_1_bn[0][0]
conv4_block9_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block9_1_relu[0][0]
conv4_block9_concat(Concatenate)	(None, 14, 14, 536)	0	conv4_block8_concat[0][0] conv4_block9_2_conv[0][0]
conv4_block10_0_bn(BatchNormalization)	(None, 14, 14, 536)	2,144	conv4_block9_concat[0][0]
conv4_block10_0_relu(Activation)	(None, 14, 14, 536)	0	conv4_block10_0_bn[0][0]
conv4_block10_1_conv(Conv2D)	(None, 14, 14, 128)	68,608	conv4_block10_0_relu[0][0]
conv4_block10_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block10_1_conv[0][0]
conv4_block10_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block10_1_bn[0][0]
conv4_block10_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block10_1_relu[0][0]
conv4_block10_concat(Concatenate)	(None, 14, 14, 568)	0	conv4_block9_concat[0][0] conv4_block10_2_conv[0][0]
conv4_block11_0_bn(BatchNormalization)	(None, 14, 14, 568)	2,272	conv4_block10_concat[0][0]
conv4_block11_0_relu(Activation)	(None, 14, 14, 568)	0	conv4_block11_0_bn[0][0]
conv4_block11_1_conv(Conv2D)	(None, 14, 14, 128)	72,704	conv4_block11_0_relu[0][0]
conv4_block11_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block11_1_conv[0][0]
conv4_block11_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block11_1_bn[0][0]
conv4_block11_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block11_1_relu[0][0]
conv4_block11_concat(Concatenate)	(None, 14, 14, 600)	0	conv4_block10_concat[0][0] conv4_block11_2_conv[0][0]
conv4_block12_0_bn(BatchNormalization)	(None, 14, 14, 600)	2,400	conv4_block11_concat[0][0]
conv4_block12_0_relu(Activation)	(None, 14, 14, 600)	0	conv4_block12_0_bn[0][0]
conv4_block12_1_conv(Conv2D)	(None, 14, 14, 128)	76,800	conv4_block12_0_relu[0][0]
conv4_block12_1_bn(BatchNormalization)	(None, 14, 14, 128)	512	conv4_block12_1_conv[0][0]
conv4_block12_1_relu(Activation)	(None, 14, 14, 128)	0	conv4_block12_1_bn[0][0]
conv4_block12_2_conv(Conv2D)	(None, 14, 14, 32)	36,864	conv4_block12_1_relu[0][0]
conv4_block12_concat(Concatenate)	(None, 14, 14, 632)	0	conv4_block11_concat[0][0] conv4_block12_2_conv[0][0]
pool4_bn(BatchNormalization)	(None, 14, 14, 632)	2,528	conv4_block12_concat[0][0]
pool4_relu(Activation)	(None, 14, 14, 632)	0	pool4_bn[0][0]
pool4_conv(Conv2D)	(None, 14, 14, 316)	199,712	pool4_relu[0][0]
pool4_drop(Dropout)	(None, 14, 14, 316)	0	pool4_conv[0][0]
pool4_pool(AveragePooling2D)	(None, 7, 7, 316)	0	pool4_drop[0][0]
conv5_block1_0_bn(BatchNormalization)	(None, 7, 7, 316)	1,264	pool4_pool[0][0]
conv5_block1_0_relu(Activation)	(None, 7, 7, 316)	0	conv5_block1_

```
class Mercedesbenzfactory(object,metaclass=Cached):
    _cache = {}
    _lock = threadlock()
    _initialized = False
    _instance = None
    def __init__(self, *args, **kwargs):
        if not hasattr(self, 'factory'):
            self.factory = Factory()
        if not self._initialized:
            super().__init__(*args, **kwargs)
            self._initialized = True
        self.logger = getlogger(__name__)
        self.logger.info('Creating Mercedesbenzfactory')
    def create_sprinter(self, *args, **kwargs):
        '''Create Sprinter using the factory.'''
        sprinter = self.factory.get_object('Sprinter', *args, **kw...
        self.logger.info('Created %s', sprinter)
        return sprinter
    def __str__(self):
        return 'Mercedesbenzfactory'
    __repr__ = __str__

MERCEDES = Mercedesbenzfactory()

# Create the factory
factory = maker.Factory()
factory.debug(1)
factory.verbose(True)
factory.load('foo.conf')
sprinter = factory.get('Sprinter')
sprinter = factory.get('Sprinter', 'diesel')
sprinter = factory.get('Sprinter', 'diesel', 5000)

# Evaluate so that we'll
s1, s2, s3 = *s1, *s2

for name in *keys:
    print('%s=%r' % (name, ...))
    if name.upper().startswith('DEBUG'):
        print('%s=%r' % (name, ...))
    elif isinstance(..., basestring):
        print('%s=%r' % (name, ...))
    else:
        print('%s=%r' % (name, ...))
print('Sprinter=%r' % (sprinter,))
print('Sprinter.__dict__=%r' % (sprinter.__dict__,))
print('Sprinter.__class__=%r' % (sprinter.__class__,))
print('Sprinter.__bases__=%r' % (sprinter.__bases__,))
print('Sprinter.__mro__=%r' % (sprinter.__mro__,))
print('Sprinter.__subclasses__()=%r' % (sprinter.__subclasses__(),))
print('Sprinter.__doc__=%r' % (sprinter.__doc__,))
print('Sprinter.__module__=%r' % (sprinter.__module__,))
print('Sprinter.__weakref__=%r' % (sprinter.__weakref__,))
print('Sprinter.__getattr__=%r' % (sprinter.__getattr__,))
print('Sprinter.__setattr__=%r' % (sprinter.__setattr__,))
print('Sprinter.__delattr__=%r' % (sprinter.__delattr__,))
print('Sprinter.__getattrib...
```

Model: `Transformer_2`

Input Tokens	Output Tokens	Parse #	Connected to
image_input	Input, TOK, EMB, POS, ATT, FFN, OUT	0	
image_input	Input, TOK, EMB, POS, ATT, FFN, OUT	900	image_input/OUT
image_input/embedding	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image/OUT
image_input/embedding/Dense	Input, TOK, EMB, POS, ATT, FFN, OUT	0	image_input/embedding
image_input/embedding/Dense/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	0	image_input/embedding/Dense
image_input/embedding/Dense/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	0	image_input/embedding/Dense
image_input/embedding/Dense/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/kernel
image_input/embedding/Dense/output/Relu	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output
image_input/embedding/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu
image_input/embedding/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu
image_input/embedding/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/kernel
image_input/embedding/Dense/output/Relu/output/Dropout	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output
image_input/embedding/Dense/output/Relu/output/Dropout/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/bias
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/bias
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/bias
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/bias
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/kernel
image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output	Input, TOK, EMB, POS, ATT, FFN, OUT	76	image_input/embedding/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu/output/Dropout/output/Dense/output/Relu

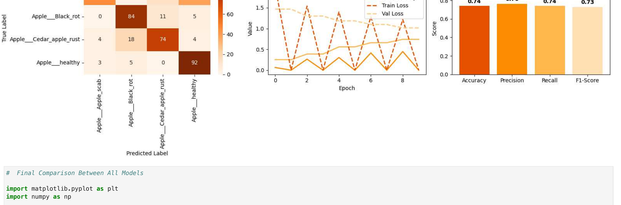
concatenate_2 (Concatenate)	(None, 5, 5, 768)	0	activation_82[0]… activation_83[0]…
activation_84 (Activation)	(None, 5, 5, 192)	0	batch_normalizat…
mixed9 (Concatenate)	(None, 5, 5, 2048)	0	activation_76[0]… mixed9_0[0][0], concatenate_2[0]… activation_84[0]…
conv2d_89 (Conv2D)	(None, 5, 5, 448)	917,504	mixed9[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 448)	1,344	conv2d_89[0][0]
activation_89 (Activation)	(None, 5, 5, 448)	0	batch_normalizat…
conv2d_86 (Conv2D)	(None, 5, 5, 384)	786,432	mixed9[0][0]
conv2d_90 (Conv2D)	(None, 5, 5, 384)	1,548,288	activation_89[0]…
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 384)	1,152	conv2d_86[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 384)	1,152	conv2d_90[0][0]
activation_86 (Activation)	(None, 5, 5, 384)	0	batch_normalizat…
activation_90 (Activation)	(None, 5, 5, 384)	0	batch_normalizat…
conv2d_87 (Conv2D)	(None, 5, 5, 384)	442,368	activation_86[0]…
conv2d_88 (Conv2D)	(None, 5, 5, 384)	442,368	activation_86[0]…
conv2d_91 (Conv2D)	(None, 5, 5, 384)	442,368	activation_90[0]…
conv2d_92 (Conv2D)	(None, 5, 5, 384)	442,368	activation_90[0]…
average_pooling2d_8 (AveragePooling2D)	(None, 5, 5, 2048)	0	mixed9[0][0]
conv2d_85 (Conv2D)	(None, 5, 5, 320)	655,360	mixed9[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 384)	1,152	conv2d_87[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 384)	1,152	conv2d_88[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 384)	1,152	conv2d_91[0][0]
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 384)	1,152	conv2d_92[0][0]
conv2d_93 (Conv2D)	(None, 5, 5, 192)	393,216	average_pooling2…
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 320)	960	conv2d_85[0][0]
activation_87 (Activation)	(None, 5, 5, 384)	0	batch_normalizat…
activation_88 (Activation)	(None, 5, 5, 384)	0	batch_normalizat…
activation_91 (Activation)	(None, 5, 5, 384)	0	batch_normalizat…
activation_92 (Activation)	(None, 5, 5, 384)	0	batch_normalizat…
batch_normalizatio… (BatchNormalizatio…	(None, 5, 5, 192)	576	conv2d_93[0][0]
activation_85 (Activation)	(None, 5, 5, 320)	0	batch_normalizat…
mixed9_1 (Concatenate)	(None, 5, 5, 768)	0	activation_87[0]… activation_88[0]…
concatenate_3 (Concatenate)	(None, 5, 5, 768)	0	activation_91[0]… activation_92[0]…
activation_93 (Activation)	(None, 5, 5, 192)	0	batch_normalizat…
mixed10 (Concatenate)	(None, 5, 5, 2048)	0	activation_85[0]… mixed9_1[0][0], concatenate_3[0]… activation_93[0]…
global_average_poo… (GlobalAveragePool…	(None, 2048)	0	mixed10[0][0]
weather_input (InputLayer)	(None, 7)	0	–
dense_10 (Dense)	(None, 128)	262,272	global_average_p…
dense_11 (Dense)	(None, 32)	256	weather_input[0]…
dropout_4 (Dropout)	(None, 128)	0	dense_10[0][0]
dense_12 (Dense)	(None, 16)	528	dense_11[0][0]
concatenate_4 (Concatenate)	(None, 144)	0	dropout_4[0][0], dense_12[0][0]
dense_13 (Dense)	(None, 64)	9,280	concatenate_4[0]…
dropout_5 (Dropout)	(None, 64)	0	dense_13[0][0]
dense_14 (Dense)	(None, 4)	260	dropout_5[0][0]

Total params: 22,075,380 (84.21 MB)
Trainable params: 272,596 (1.04 MB)
Non-trainable params: 21,802,784 (83.17 MB)

```
Epoch 1/10
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 878ms/step - accuracy: 0.0429 - loss: 2.1564 - val_precision: 0.2149 - val_recall: 0.2525 - val_f1: 0.1740
13/13 ━━━━━━━━━━━━━━━━━━━━ 54s 3s/step - accuracy: 0.0650 - loss: 2.0186 - val_accuracy: 0.2525 - val_loss: 1.4653 - learning_rate: 1.0000e-04 - val_precision: 0.2149 -
val_recall: 0.2525 - val_f1: 0.1740
Epoch 2/10
- val_precision: 0.2149 - val_recall: 0.2525 - val_f1: 0.1740
13/13 ━━━━━━━━━━━━━━━━━━━━ 24s 2s/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.2525 - val_loss: 1.4653 - learning_rate: 1.0000e-04 - val_precision:
0.2149 - val_recall: 0.2525 - val_f1: 0.1740
Epoch 3/10
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 846ms/step - accuracy: 0.3045 - loss: 1.4653 - val_precision: 0.3814 - val_recall: 0.3875 - val_f1: 0.3419
13/13 ━━━━━━━━━━━━━━━━━━━━ 35s 3s/step - accuracy: 0.2658 - loss: 1.5393 - val_accuracy: 0.3875 - val_loss: 1.2993 - learning_rate: 1.0000e-04 - val_precision: 0.3814 -
val_recall: 0.3875 - val_f1: 0.3419
Epoch 4/10
- val_precision: 0.3814 - val_recall: 0.3875 - val_f1: 0.3419
13/13 ━━━━━━━━━━━━━━━━━━━━ 23s 2s/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.3875 - val_loss: 1.2993 - learning_rate: 1.0000e-04 - val_precision:
0.3814 - val_recall: 0.3875 - val_f1: 0.3419
Epoch 5/10
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 869ms/step - accuracy: 0.3049 - loss: 1.3943 - val_precision: 0.5613 - val_recall: 0.5600 - val_f1: 0.5439
13/13 ━━━━━━━━━━━━━━━━━━━━ 35s 3s/step - accuracy: 0.3050 - loss: 1.3982 - val_accuracy: 0.5600 - val_loss: 1.1836 - learning_rate: 1.0000e-04 - val_precision: 0.5613 -
val_recall: 0.5600 - val_f1: 0.5439
Epoch 6/10
- val_precision: 0.5613 - val_recall: 0.5600 - val_f1: 0.5439
13/13 ━━━━━━━━━━━━━━━━━━━━ 23s 2s/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.5600 - val_loss: 1.1836 - learning_rate: 1.0000e-04 - val_precision:
0.5613 - val_recall: 0.5600 - val_f1: 0.5439
Epoch 7/10
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 861ms/step - accuracy: 0.3331 - loss: 1.3657 - val_precision: 0.7035 - val_recall: 0.6600 - val_f1: 0.6391
13/13 ━━━━━━━━━━━━━━━━━━━━ 36s 3s/step - accuracy: 0.4150 - loss: 1.2796 - val_accuracy: 0.6600 - val_loss: 1.0970 - learning_rate: 1.0000e-04 - val_precision: 0.7035 -
val_recall: 0.6600 - val_f1: 0.6391
Epoch 8/10
- val_precision: 0.7035 - val_recall: 0.6600 - val_f1: 0.6391
13/13 ━━━━━━━━━━━━━━━━━━━━ 23s 2s/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.6600 - val_loss: 1.0970 - learning_rate: 1.0000e-04 - val_precision:
0.7035 - val_recall: 0.6600 - val_f1: 0.6391
Epoch 9/10
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 865ms/step - accuracy: 0.3577 - loss: 1.3023 - val_precision: 0.7627 - val_recall: 0.7400 - val_f1: 0.7298
13/13 ━━━━━━━━━━━━━━━━━━━━ 36s 3s/step - accuracy: 0.4475 - loss: 1.2162 - val_accuracy: 0.7400 - val_loss: 1.0151 - learning_rate: 1.0000e-04 - val_precision: 0.7627 -
val_recall: 0.7400 - val_f1: 0.7298
Epoch 10/10
- val_precision: 0.7627 - val_recall: 0.7400 - val_f1: 0.7298
13/13 ━━━━━━━━━━━━━━━━━━━━ 24s 2s/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.7400 - val_loss: 1.0151 - learning_rate: 1.0000e-04 - val_precision:
0.7627 - val_recall: 0.7400 - val_f1: 0.7298

Test Accuracy: 0.7400
Test Precision: 0.7627
Test Recall: 0.7400
Test F1-Score: 0.7298

Classification Report:
                     precision    recall  f1-score   support

      Apple___Apple_scab       0.87      0.46      0.60       100
   Apple___Black_rot       0.77      0.84      0.80       100
Apple___Cedar_apple_rust       0.76      0.74      0.75       100
     Apple___healthy       0.66      0.92      0.77       100

            accuracy                           0.74       400
           macro avg       0.76      0.74      0.73       400
        weighted avg       0.76      0.74      0.73       400
```

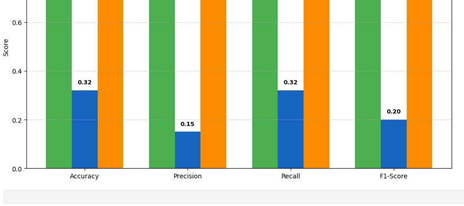
In [15]:

```
# Visualizations for InceptionV3 Multimodal Model

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
import numpy as np

# 1. Confusion Matrix
cm = confusion_matrix(y_true, y_pred)
class_names = list(test_gen.class_indices.keys())

plt.figure(figsize=(16,5))

# ---- Confusion Matrix ----
plt.subplot(1,3,1)
sns.heatmap(cm, annot=True, fmt="d", cmap="Oranges",
            xticklabels=class_names, yticklabels=class_names)
plt.title("InceptionV3 - Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")

# 2. Accuracy & Loss Curves
plt.subplot(1,3,2)
plt.plot(history.history["accuracy"], label="Train Accuracy", linewidth=2, color="#FBBC00")
plt.plot(history.history["val_accuracy"], label="Val Accuracy", linewidth=2, color="#FFB74D")
plt.plot(history.history["loss"], label="Train Loss", linestyle="--", linewidth=2, color="#E65100")
plt.plot(history.history["val_loss"], label="Val Loss", linestyle="--", linewidth=2, color="#FFCC80")
plt.legend()
plt.title("InceptionV3 - Accuracy & Loss over Epochs")
plt.xlabel("Epoch")
plt.ylabel("Value")

# 3. Performance Metrics Bar Chart
metrics = ["Accuracy", "Precision", "Recall", "F1-Score"]
values = [acc, prec, rec, f1]

plt.subplot(1,3,3)
bars = plt.bar(metrics, values, color=["#E65100", "#FBBC00", "#FFB74D", "#FFE082"])
plt.ylim(0, 1)
plt.title("InceptionV3 - Model Performance Metrics")
plt.ylabel("Score")

for bar, val in zip(bars, values):
    plt.text(bar.get_x() + bar.get_width()/2, val + 0.02, f"{val:.2f}",
             ha='center', va='bottom', fontsize=10, fontweight='bold')

plt.tight_layout()
plt.show()
```



In [17]:

```
# Final Comparison Between All Models

import matplotlib.pyplot as plt
import numpy as np

# Insert your final metrics for each model

mobilenet_metrics = [0.7825, 0.7858, 0.7825, 0.7805]
resnet_metrics    = [0.3200, 0.1502, 0.3200, 0.2008]
inception_metrics = [0.7400, 0.7627, 0.7400, 0.7298]

metrics_names = ["Accuracy", "Precision", "Recall", "F1-Score"]

# Create grouped bar chart
x = np.arange(len(metrics_names))
width = 0.25

plt.figure(figsize=(10,6))
plt.bar(x - width, mobilenet_metrics, width, label="MobileNetV2", color="#4CAF50")
plt.bar(x, resnet_metrics, width, label="ResNet50", color="#1565C0")
plt.bar(x + width, inception_metrics, width, label="InceptionV3", color="#FBBC00")

# Customize chart
plt.xticks(x, metrics_names)
plt.ylim(0, 1)
plt.ylabel("Score")
plt.title("📊 Model Performance Comparison (Apple Disease Detection)")
plt.legend()
plt.grid(axis="y", linestyle="--", alpha=0.6)

# Annotate bars with values
for i, values in enumerate([mobilenet_metrics, resnet_metrics, inception_metrics]):
    for j, val in enumerate(values):
        plt.text(j + (i-1)*width, val + 0.02, f"{val:.2f}",
                 ha='center', va='bottom', fontsize=9, fontweight='bold')

plt.tight_layout()
plt.show()
```



In [ ]: