endg511project

April 14, 2023

1 Dashcam Vehicle Identification Using CNNs and Model Compression

Report by: Ryan Baker (10111414) and Nathalie Drzewiecki (30090300)

1.1 Summary

In this project, we utilize convolutional neural networks (CNNs) to complete a car make-model classification task based off single RGB images. Initially, the intent was to compare the accuracy and performance of a traditional large neural network, AlexNet, to a more novel and compressed network, MobileNet. Unfortunately, the project proceeded with difficulty, and led to explorations of the dataset, hyper-parameters and models in an effort to reach an accuracy of over 90%. After much work in this area, the final models only reached a validation accuracy of around 50%. Finally, despite the lack of performance in the final models, the effects of iterative pruning and quantization were explored with compression applied to the MobileNet model. In conclusion, we identified the major obstacle in car make-model classfication to be the dataset itself. While a large and comprehensive dataset seems ideal, the effects of unbalanced classes with varying numbers of images as well as the granular classification of makes and models by year lead to various challenges within the project.

1.2 Problem Statement and Use Case

With the common adoption of dashcam devices in vehicles, thousands of hours of road footage are now easy to access, but without any practical use beyond immediate security concerns (eg. recording a road accident). The implementation of a ML model gives this footage additional practical use. Therefore, the purpose of this study is to explore the feasibility of a mobile processing unit (such as a raspberry pi) that could process dashcam footage. A large data set ("Vehicle Make and Model Recognition Database"; abbrev. "VMMRdb") will be used to first attempt to develop an accurate make-model identifier. Once an accurate model is developed, we will attempt to compress the model to a practical size/complexity for edge deployment. Potential applications of the model could be statistics about vehicles, dates/times of encounters, and may be useful in the policing, surveillance, security, or insurance fields.

1.3 Dataset

The data base chosen is the VMMRdb [1], comprised of 291,752 images covering models manufactured between 1950 and 2016. It includes different imaging devices, various view angles, random alignment within the image, and images with irrelevant backgrounds. The data comes from 712 areas within the United States and covers "all 412 subdomains". The dataset was originally created for developing robust ML models in real life scenarios for traffic surveillance. This data set was chosen to to its large diversity of images, and most accurately represents the variability of vehicles a make/model/year classifier could encounter on the road, but also the frequency (eg. common models like

A downloadable version of the dataset was found at https://www.kaggle.com/datasets/abhishektyagi001/vehicle-make-model-recognition-dataset-vmmrdb

1.4 Proposed Methodology

We propose a solution utilizing a Convolutional Neural Network, which will classify car models based off single RGB images. The size of the model will be optimized in terms of number of layers, kernel sizes, stride sizes, and other parameters. Additionally, the model will be pruned and quantized for size reduction.

Two architectures will be explored and compared: AlexNet

1.5 Project Code and Discussion

1.6 VMMRdb and AlexNet

1.6.1 Mount Colab Drive

[ ]:
```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

1.6.2 Extracting VMMRdb Dataset

[ ]:
```
import os
import shutil
shutil.unpack_archive("/content/drive/MyDrive/endg511project/VMMRdb.zip", "/
  ↪content/")
```

Delete empty folders

[ ]:
```
!find /content/VMMRdb/ -size  0 -print -delete
```

[ ]:
```
num_classes = len(os.listdir("/content/VMMRdb/"))
print(num_classes)
```

```
8174
```

1.6.3 Train AlexNet with VMMRdb Dataset

[ ]:
```
import tensorflow as tf
from matplotlib import pyplot as plt
from PIL import Image
import functools

DIRPATH = '/content/VMMRdb/'
```

```
num_classes = len(os.listdir(DIRPATH))
print(f"Number of classes: {num_classes}")
```

```
Number of classes: 8174
```

[ ]:

```
train_alexvmmrdb, test_alexvmmrdb = tf.keras.utils.image_dataset_from_directory(
    DIRPATH,
    validation_split=0.2,
    subset="both",
    seed=123,
    image_size=[227,227],
    label_mode='int')
```

```
Found 277231 files belonging to 8174 classes.
Using 221785 files for training.
Using 55446 files for validation.
```

[ ]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import␣
 ↪Conv2D,MaxPool2D,Dense,Flatten,Dropout,Input, AveragePooling2D,␣
 ↪Activation,Conv2D, MaxPooling2D, BatchNormalization,Concatenate
alexmodel = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(filters=96, kernel_size=(11,11), strides=(4,4),␣
 ↪activation='relu', input_shape=(227,227,3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Conv2D(filters=256, kernel_size=(5,5), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(filters=256, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])
```

```python
# printing the model summary
alexmodel.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 55, 55, 96)        34944

 batch_normalization (BatchN  (None, 55, 55, 96)       384
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 27, 27, 96)       0
 )

 conv2d_1 (Conv2D)           (None, 27, 27, 256)       614656

 batch_normalization_1 (Batc  (None, 27, 27, 256)      1024
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 13, 13, 256)      0
 2D)

 conv2d_2 (Conv2D)           (None, 13, 13, 384)       885120

 batch_normalization_2 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 conv2d_3 (Conv2D)           (None, 13, 13, 384)       1327488

 batch_normalization_3 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 conv2d_4 (Conv2D)           (None, 13, 13, 256)       884992

 batch_normalization_4 (Batc  (None, 13, 13, 256)      1024
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 flatten (Flatten)           (None, 9216)              0

 dense (Dense)               (None, 4096)              37752832

 dropout (Dropout)           (None, 4096)              0
 dense_1 (Dense)             (None, 4096)              16781312

 dropout_1 (Dropout)         (None, 4096)              0

 dense_2 (Dense)             (None, 8174)              33488878

=================================================================
Total params: 91,775,726
Trainable params: 91,772,974
Non-trainable params: 2,752
_________________________________________________________________
```

[ ]:

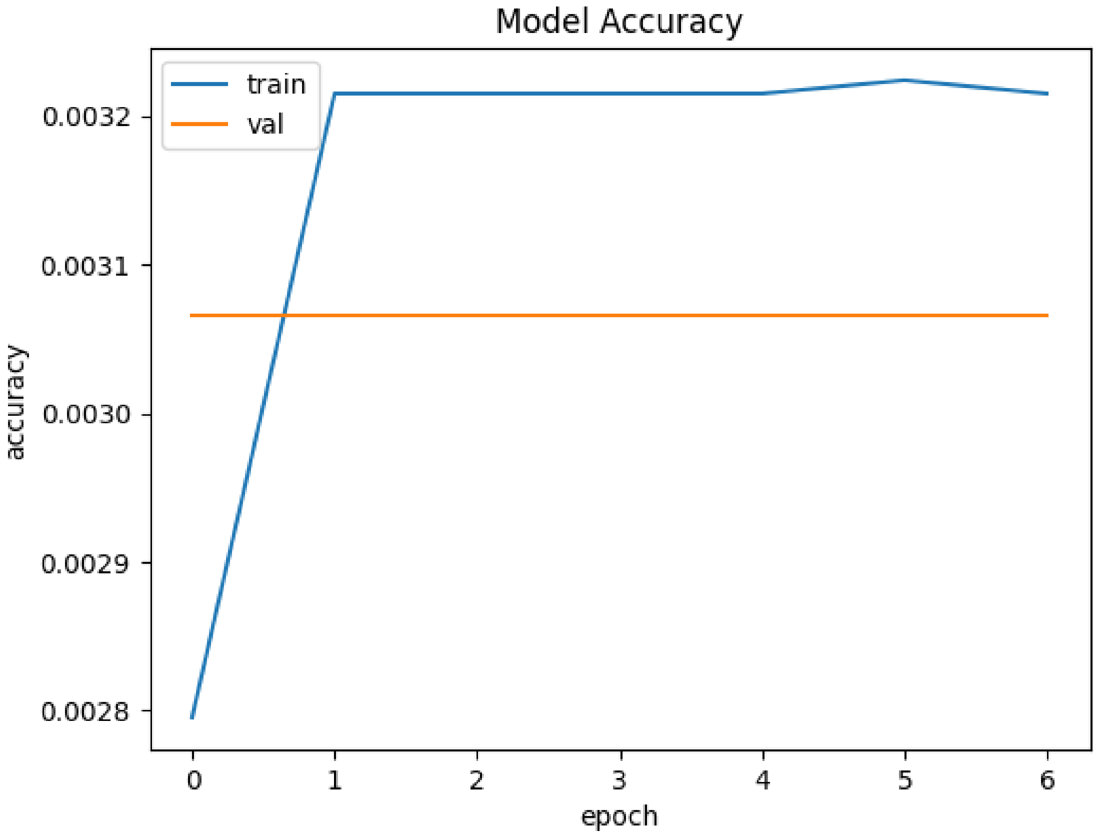
```python
#Call back 1:
base_learning_rate = 0.001
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='accuracy', min_delta=0.005, patience=6, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

#call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1,callback_2]

#compiling our Model for datasert
alexmodel.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = alexmodel.fit(train_alexvmmrdb, epochs=30,␣
 ↪validation_data=test_alexvmmrdb,callbacks=callback_list)
```

```
Epoch 1/30
6931/6931 [==============================] - 147s 19ms/step - loss: 8.5228 -
accuracy: 0.0028 - val_loss: 8.2557 - val_accuracy: 0.0031 - lr: 0.0010
Epoch 2/30
6931/6931 [==============================] - 129s 19ms/step - loss: 8.1687 -
accuracy: 0.0032 - val_loss: 8.1656 - val_accuracy: 0.0031 - lr: 0.0010
Epoch 3/30
6931/6931 [==============================] - 131s 19ms/step - loss: 8.1117 -
accuracy: 0.0032 - val_loss: 8.1536 - val_accuracy: 0.0031 - lr: 0.0010
```

```
Epoch 4/30
6931/6931 [==============================] - 130s 19ms/step - loss: 8.0981 -
accuracy: 0.0032 - val_loss: 8.1534 - val_accuracy: 0.0031 - lr: 0.0010
Epoch 5/30
6931/6931 [==============================] - 130s 19ms/step - loss: 8.0950 -
accuracy: 0.0032 - val_loss: 8.1552 - val_accuracy: 0.0031 - lr: 0.0010
Epoch 6/30
6931/6931 [==============================] - 130s 19ms/step - loss: 8.0942 -
accuracy: 0.0032 - val_loss: 8.1571 - val_accuracy: 0.0031 - lr: 0.0010
Epoch 7/30
6931/6931 [==============================] - 130s 19ms/step - loss: 8.0946 -
accuracy: 0.0032 - val_loss: 8.1590 - val_accuracy: 0.0031 - lr: 0.0010
```

[ ]:

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



Given the poor performance of this model, we will try to use `SparseTopKCategoricalAccuracy` as a metric instead of accuracy. Given the large number of classes, this may help improve this initial model's performance by relaxing the .

Determining how many 'k' classes to use

[ ]:

```
import pandas as pd
data = []
for dir in os.listdir('./VMMRdb/'):
  dir_data = dir.split("_")[:2]
  data.append(dir_data)

df = pd.DataFrame(data)
df = df.groupby([0, 1]).size().reset_index(name='frequency')
print(df)
print(f'Average number of classes for unique make-models: {df["frequency"].
  ↪mean()}')
print(f'Standard deviation of number of classes for unique make-models:␣
  ↪{df["frequency"].std()}')
```

```
          0        1  frequency
0     acura       cl          6
1     acura       el          3
2     acura      ilx          1
3     acura  integra         16
4     acura   legend         11
..      …        …          …
959   volvo     xc60          3
960   volvo     xc70         13
961   volvo     xc90         10
962  willys     cj2a          1
963  willys     cj3b          1

[964 rows x 3 columns]
Average number of classes for unique make-models: 8.479253112033195
Standard deviation of number of classes for unique make-models:
10.840416331158888
```

Given the large standard deviation of classes, using two standard deviations should accomodate more of each unique make/model than using just the mean for a k value (therefore, k=22 will be used).

[ ]:

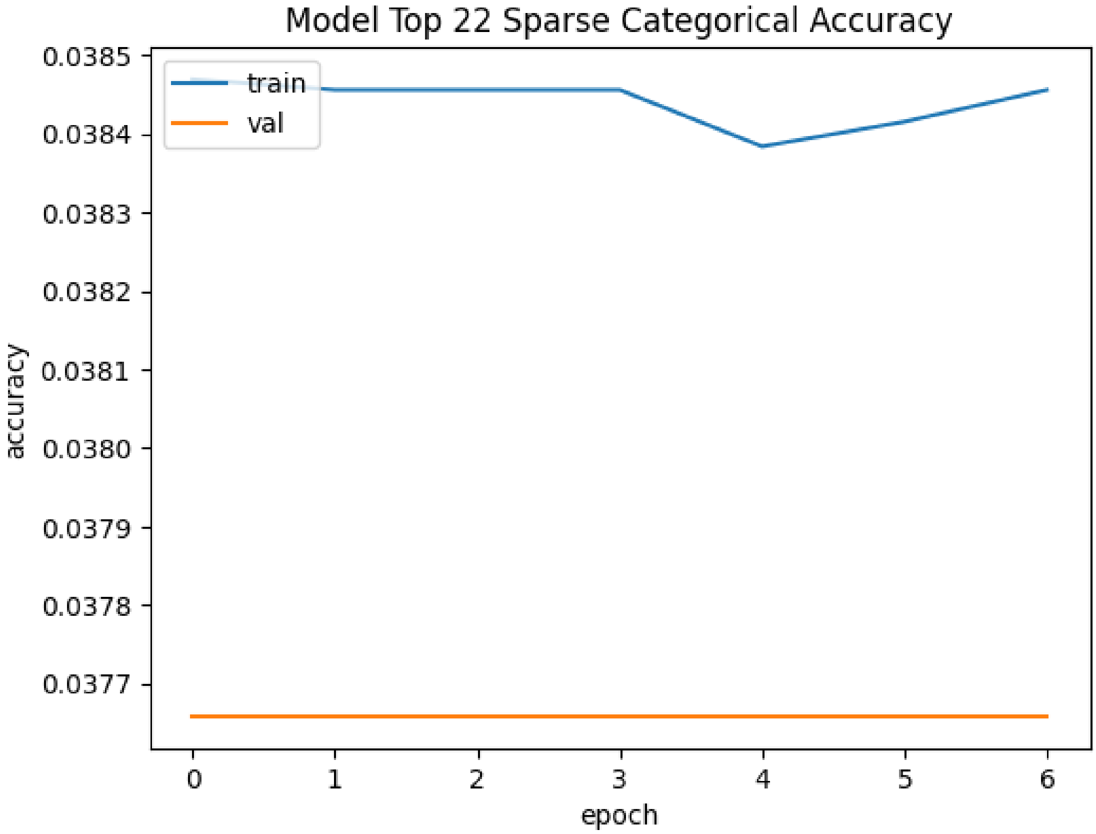
```
#Call back 1:
base_learning_rate = 0.001
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='sparse_top_k_categorical_accuracy', min_delta=0.005, patience=6,␣
  ↪verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)
```

```
#call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1,callback_2]

#compiling our Model for datasert
alexmodel.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=[tf.keras.metrics.SparseTopKCategoricalAccuracy(k=22)])

# training the model and saving the model components history to history variable
history = alexmodel.fit(train_alexvmmrdb, epochs=30,␣
 ↪validation_data=test_alexvmmrdb,callbacks=callback_list)
```

```
Epoch 1/30
6931/6931 [==============================] - 130s 18ms/step - loss: 8.1708 -
sparse_top_k_categorical_accuracy: 0.0385 - val_loss: 8.1769 -
val_sparse_top_k_categorical_accuracy: 0.0377 - lr: 0.0010
Epoch 2/30
6931/6931 [==============================] - 124s 18ms/step - loss: 8.1200 -
sparse_top_k_categorical_accuracy: 0.0385 - val_loss: 8.1694 -
val_sparse_top_k_categorical_accuracy: 0.0377 - lr: 0.0010
Epoch 3/30
6931/6931 [==============================] - 124s 18ms/step - loss: 8.1089 -
sparse_top_k_categorical_accuracy: 0.0385 - val_loss: 8.1685 -
val_sparse_top_k_categorical_accuracy: 0.0377 - lr: 0.0010
Epoch 4/30
6931/6931 [==============================] - 127s 18ms/step - loss: 8.1054 -
sparse_top_k_categorical_accuracy: 0.0385 - val_loss: 8.1687 -
val_sparse_top_k_categorical_accuracy: 0.0377 - lr: 0.0010
Epoch 5/30
6931/6931 [==============================] - 127s 18ms/step - loss: 8.1040 -
sparse_top_k_categorical_accuracy: 0.0384 - val_loss: 8.1691 -
val_sparse_top_k_categorical_accuracy: 0.0377 - lr: 0.0010
Epoch 6/30
6931/6931 [==============================] - 127s 18ms/step - loss: 8.1030 -
sparse_top_k_categorical_accuracy: 0.0384 - val_loss: 8.1696 -
val_sparse_top_k_categorical_accuracy: 0.0377 - lr: 0.0010
Epoch 7/30
6931/6931 [==============================] - 127s 18ms/step - loss: 8.1026 -
sparse_top_k_categorical_accuracy: 0.0385 - val_loss: 8.1700 -
```

```
val_sparse_top_k_categorical_accuracy: 0.0377 - lr: 0.0010
```

[ ]:
```
plt.plot(history.history['sparse_top_k_categorical_accuracy'])
plt.plot(history.history['val_sparse_top_k_categorical_accuracy'])
plt.title('Model Top 22 Sparse Categorical Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



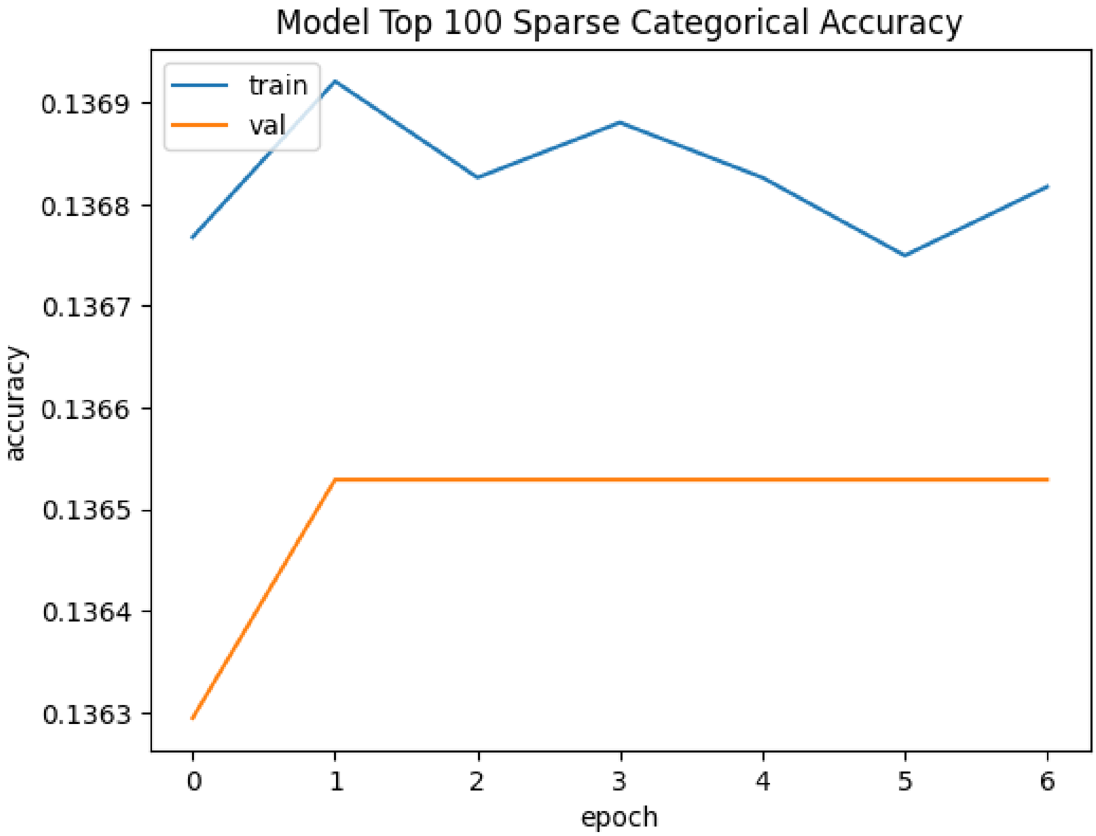
Try with k=200 to see if any decent accuracy can be obtained

[ ]:
```
#Call back 1:
base_learning_rate = 0.001
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='sparse_top_k_categorical_accuracy', min_delta=0.005, patience=6,␣
 ↪verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

#call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1,callback_2]

#compiling our Model for datasert
alexmodel.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=[tf.keras.metrics.SparseTopKCategoricalAccuracy(k=200)])

# training the model and saving the model components history to history variable
history = alexmodel.fit(train_alexvmmrdb, epochs=30,␣
 ↪validation_data=test_alexvmmrdb,callbacks=callback_list)
```

```
Epoch 1/30
6931/6931 [==============================] - 132s 18ms/step - loss: 8.1638 -
sparse_top_k_categorical_accuracy: 0.1368 - val_loss: 8.2125 -
val_sparse_top_k_categorical_accuracy: 0.1363 - lr: 0.0010
Epoch 2/30
6931/6931 [==============================] - 126s 18ms/step - loss: 8.1451 -
sparse_top_k_categorical_accuracy: 0.1369 - val_loss: 8.2063 -
val_sparse_top_k_categorical_accuracy: 0.1365 - lr: 0.0010
Epoch 3/30
6931/6931 [==============================] - 126s 18ms/step - loss: 8.1373 -
sparse_top_k_categorical_accuracy: 0.1368 - val_loss: 8.2020 -
val_sparse_top_k_categorical_accuracy: 0.1365 - lr: 0.0010
Epoch 4/30
6931/6931 [==============================] - 126s 18ms/step - loss: 8.1318 -
sparse_top_k_categorical_accuracy: 0.1369 - val_loss: 8.1989 -
val_sparse_top_k_categorical_accuracy: 0.1365 - lr: 0.0010
Epoch 5/30
6931/6931 [==============================] - 127s 18ms/step - loss: 8.1279 -
sparse_top_k_categorical_accuracy: 0.1368 - val_loss: 8.1963 -
val_sparse_top_k_categorical_accuracy: 0.1365 - lr: 0.0010
Epoch 6/30
6931/6931 [==============================] - 127s 18ms/step - loss: 8.1247 -
sparse_top_k_categorical_accuracy: 0.1367 - val_loss: 8.1942 -
val_sparse_top_k_categorical_accuracy: 0.1365 - lr: 0.0010
Epoch 7/30
6931/6931 [==============================] - 126s 18ms/step - loss: 8.1228 -
sparse_top_k_categorical_accuracy: 0.1368 - val_loss: 8.1924 -
val_sparse_top_k_categorical_accuracy: 0.1365 - lr: 0.0010
```

[ ]:

```
plt.plot(history.history['sparse_top_k_categorical_accuracy'])
plt.plot(history.history['val_sparse_top_k_categorical_accuracy'])
plt.title('Model Top 200 Sparse Categorical Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



Since we are still only trying to get a good performing **initial** model what we can then prune/quantize, given the poor performance thus far, the large number of classes, and the few number of images per class, it may be better to combine classes and attempt to retrain.

The script below combines make/model/year classes into make/model classes, and then deletes combined classes that have fewer than 100 images. This should reduce the number of classes by approximately 20x, and also increase the number of images by approximately 10x. Additionally, the year-to-year changes of the same make/model is assumed to be insignificant, so this should substantially improve the model performance when retraining.

[ ]:

```
def make_dir(path_to_dir):
    if os.path.exists(path_to_dir):
        return path_to_dir
    else:
        os.mkdir(path_to_dir)
        return path_to_dir

OLD_DIR = '/content/VMMRdb/'
NEW_DIR = make_dir('/content/VMMRdb_reduced_classes/')

for i, dir in enumerate(os.listdir(OLD_DIR)):
    new_file_dir = make_dir(NEW_DIR+"_".join(dir.split("_")[:2]))
    for j, file in enumerate(os.listdir(OLD_DIR+'/'+dir)):
        old_file_loc = OLD_DIR+dir+'/'+file
        new_file_loc = new_file_dir + '/' + file
        shutil.move(old_file_loc, new_file_loc)

# delete directories with fewer than 100 files
min_file_count = 100

for dir in os.listdir(NEW_DIR):
    subdir = NEW_DIR+'/'+dir
    file_count = len(os.listdir(subdir))
    if file_count < min_file_count:
        shutil.rmtree(subdir)

# delete old directory of empty folders
shutil.rmtree(OLD_DIR)
```

[ ]:

```
DIRPATH = NEW_DIR
num_classes = len(os.listdir(DIRPATH))
print(num_classes)
```

```
363
```

1.6.4 Train AlexNet with VMMRdb Dataset After Combining 'Year' Classes

[ ]:

```
train_alexvmmrdb, test_alexvmmrdb = tf.keras.utils.image_dataset_from_directory(
    DIRPATH,
    validation_split=0.2,
    subset="both",
    seed=123,
    image_size=[227,227],
    label_mode='int')
```

```
Found 262640 files belonging to 363 classes.
Using 210112 files for training.
Using 52528 files for validation.
```

[ ]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import␣
 ↪Conv2D,MaxPool2D,Dense,Flatten,Dropout,Input, AveragePooling2D,␣
 ↪Activation,Conv2D, MaxPooling2D, BatchNormalization,Concatenate
alexmodel = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(filters=96, kernel_size=(11,11), strides=(4,4),␣
 ↪activation='relu', input_shape=(227,227,3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Conv2D(filters=256, kernel_size=(5,5), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(filters=256, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])

# printing the model summary
alexmodel.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_5 (Conv2D)           (None, 55, 55, 96)        34944

 batch_normalization_5 (Batc  (None, 55, 55, 96)       384
 hNormalization)

 max_pooling2d_3 (MaxPooling  (None, 27, 27, 96)       0
 2D)

 conv2d_6 (Conv2D)           (None, 27, 27, 256)       614656
 batch_normalization_6 (Batc  (None, 27, 27, 256)      1024
 hNormalization)

 max_pooling2d_4 (MaxPooling  (None, 13, 13, 256)      0
 2D)

 conv2d_7 (Conv2D)           (None, 13, 13, 384)       885120

 batch_normalization_7 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 conv2d_8 (Conv2D)           (None, 13, 13, 384)       1327488

 batch_normalization_8 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 conv2d_9 (Conv2D)           (None, 13, 13, 256)       884992

 batch_normalization_9 (Batc  (None, 13, 13, 256)      1024
 hNormalization)

 max_pooling2d_5 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 flatten_1 (Flatten)         (None, 9216)              0

 dense_3 (Dense)             (None, 4096)              37752832

 dropout_2 (Dropout)         (None, 4096)              0

 dense_4 (Dense)             (None, 4096)              16781312

 dropout_3 (Dropout)         (None, 4096)              0

 dense_5 (Dense)             (None, 363)               1487211

=================================================================
Total params: 59,774,059
Trainable params: 59,771,307
Non-trainable params: 2,752
_________________________________________________________________
```

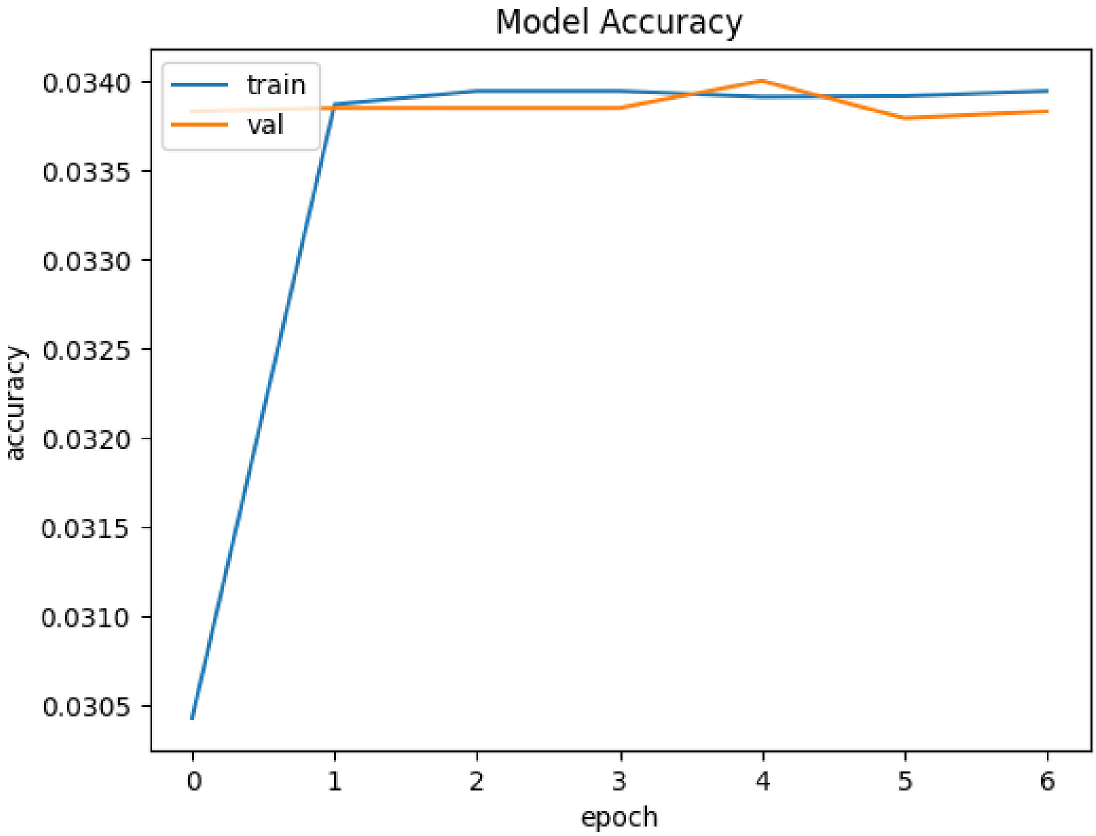
[ ]:

```
#Call back 1:
base_learning_rate = 0.001
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='accuracy', min_delta=0.005, patience=6, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

#call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1,callback_2]

#compiling our Model for datasert
alexmodel.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = alexmodel.fit(train_alexvmmrdb, epochs=30,␣
  ↪validation_data=test_alexvmmrdb,callbacks=callback_list)
```

```
Epoch 1/30
6566/6566 [==============================] - 119s 17ms/step - loss: 5.6290 -
accuracy: 0.0304 - val_loss: 5.2576 - val_accuracy: 0.0338 - lr: 0.0010
Epoch 2/30
6566/6566 [==============================] - 113s 17ms/step - loss: 5.2664 -
accuracy: 0.0339 - val_loss: 5.2565 - val_accuracy: 0.0338 - lr: 0.0010
Epoch 3/30
6566/6566 [==============================] - 113s 17ms/step - loss: 5.2575 -
accuracy: 0.0339 - val_loss: 5.2564 - val_accuracy: 0.0338 - lr: 0.0010
Epoch 4/30
6566/6566 [==============================] - 113s 17ms/step - loss: 5.2565 -
accuracy: 0.0339 - val_loss: 5.2561 - val_accuracy: 0.0338 - lr: 0.0010
Epoch 5/30
6566/6566 [==============================] - 113s 17ms/step - loss: 5.2609 -
accuracy: 0.0339 - val_loss: 16143.0947 - val_accuracy: 0.0340 - lr: 0.0010
Epoch 6/30
6566/6566 [==============================] - 113s 17ms/step - loss: 5.2568 -
accuracy: 0.0339 - val_loss: 542.7129 - val_accuracy: 0.0338 - lr: 0.0010
Epoch 7/30
6566/6566 [==============================] - 113s 17ms/step - loss: 5.2548 -
accuracy: 0.0339 - val_loss: 2510.5237 - val_accuracy: 0.0338 - lr: 0.0010
```

[ ]:

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



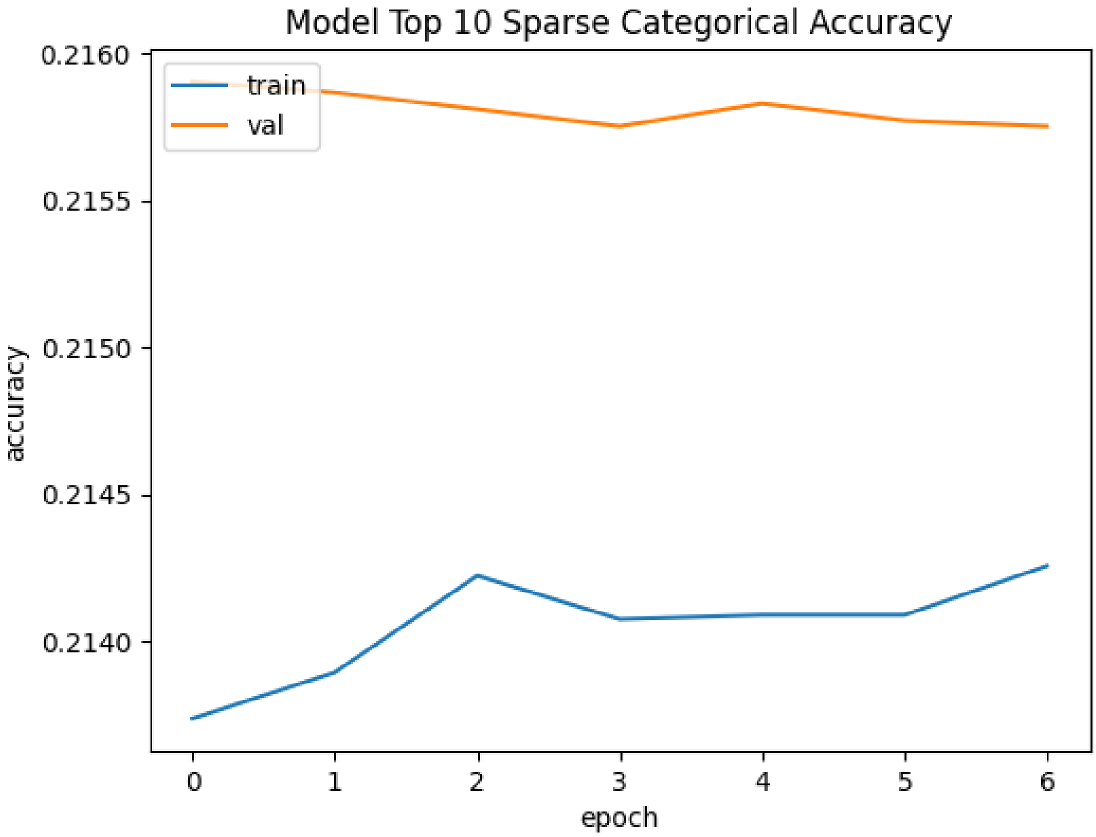
As only 3.4% accuracy was attained, using top 10 class accuracy may improve the results. *It should be noted that by using a* ***top k*** *metric with this many reduced classes, it's possible (and even likely) that a class within the 'top k' may not even be the same make of vehicle.* However, for the purposes of experimentation and in order to strive for a higher accuracy, this test was conducted.

[ ]:

```
#Call back 1:
base_learning_rate = 0.001
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='sparse_top_k_categorical_accuracy', min_delta=0.005, patience=6,␣
 ↪verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

#call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1,callback_2]

#compiling our Model for datasert
alexmodel.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=[tf.keras.metrics.SparseTopKCategoricalAccuracy(k=10)])

# training the model and saving the model components history to history variable
history = alexmodel.fit(train_alexvmmrdb, epochs=30,␣
 ↪validation_data=test_alexvmmrdb,callbacks=callback_list)
```

```
Epoch 1/30
6566/6566 [==============================] - 122s 18ms/step - loss: 5.2580 -
sparse_top_k_categorical_accuracy: 0.2137 - val_loss: 57.6710 -
val_sparse_top_k_categorical_accuracy: 0.2159 - lr: 0.0010
Epoch 2/30
6566/6566 [==============================] - 117s 18ms/step - loss: 5.2559 -
sparse_top_k_categorical_accuracy: 0.2139 - val_loss: 55.5179 -
val_sparse_top_k_categorical_accuracy: 0.2159 - lr: 0.0010
Epoch 3/30
6566/6566 [==============================] - 115s 17ms/step - loss: 5.2556 -
sparse_top_k_categorical_accuracy: 0.2142 - val_loss: 53.9903 -
val_sparse_top_k_categorical_accuracy: 0.2158 - lr: 0.0010
Epoch 4/30
6566/6566 [==============================] - 115s 17ms/step - loss: 5.2548 -
sparse_top_k_categorical_accuracy: 0.2141 - val_loss: 59.3962 -
val_sparse_top_k_categorical_accuracy: 0.2158 - lr: 0.0010
Epoch 5/30
6566/6566 [==============================] - 115s 17ms/step - loss: 5.2548 -
sparse_top_k_categorical_accuracy: 0.2141 - val_loss: 54.2885 -
val_sparse_top_k_categorical_accuracy: 0.2158 - lr: 0.0010
Epoch 6/30
6566/6566 [==============================] - 115s 17ms/step - loss: 5.2541 -
sparse_top_k_categorical_accuracy: 0.2141 - val_loss: 60.9703 -
val_sparse_top_k_categorical_accuracy: 0.2158 - lr: 0.0010
Epoch 7/30
6566/6566 [==============================] - 117s 18ms/step - loss: 5.2539 -
sparse_top_k_categorical_accuracy: 0.2143 - val_loss: 54.9004 -
val_sparse_top_k_categorical_accuracy: 0.2158 - lr: 0.0010
```

```
[ ]: plt.plot(history.history['sparse_top_k_categorical_accuracy'])
     plt.plot(history.history['val_sparse_top_k_categorical_accuracy'])
     plt.title('Model Top 10 Sparse Categorical Accuracy')
     plt.ylabel('accuracy')
     plt.xlabel('epoch')
     plt.legend(['train', 'val'], loc='upper left')
     plt.show()
```



1.6.5 Train AlexNet with VMMRdb Dataset After Combining 'Model' Classes

As a final attempt to use the AlexNet model with this data set, we will further combine the classes into make only. This will significantly impact the use case, however the purpose of this study was to investigate the feasibility of identifying vehicles in real time from dashcam footage. There could still be reasonable use cases for identifying only the make of vehicles.

If this model also has low accuracy, the error may be related to the AlexNet architecture or the images themselves. Given the high variety of images (various sources, various backgrounds, etc), it's possible that the AlexNet architecure isn't very good at identifying the key features of this data set or that the images have too much 'noise' for a CNN to identify the distinguishing features.

[ ]:

```python
def make_dir(path_to_dir):
    if os.path.exists(path_to_dir):
        return path_to_dir
    else:
        os.mkdir(path_to_dir)
        return path_to_dir

OLD_DIR = '/content/VMMRdb_reduced_classes/'
NEW_DIR = make_dir('/content/VMMRdb_make_only/')

for i, dir in enumerate(os.listdir(OLD_DIR)):
    new_file_dir = make_dir(NEW_DIR+"_".join(dir.split("_")[:1]))
    for j, file in enumerate(os.listdir(OLD_DIR+'/'+dir)):
        old_file_loc = OLD_DIR+dir+'/'+file
        new_file_loc = new_file_dir + '/' + file
        shutil.move(old_file_loc, new_file_loc)

# delete directories with fewer than 100 files
min_file_count = 100

for dir in os.listdir(NEW_DIR):
    subdir = NEW_DIR+'/'+dir
    file_count = len(os.listdir(subdir))
    if file_count < min_file_count:
        shutil.rmtree(subdir)

# delete old directory of empty folders
shutil.rmtree(OLD_DIR)
```

[ ]:

```python
DIRPATH = NEW_DIR
num_classes = len(os.listdir(DIRPATH))
print(num_classes)
```

```
41
```

[ ]:

```python
train_alexvmmrdb, test_alexvmmrdb = tf.keras.utils.image_dataset_from_directory(
    DIRPATH,
    validation_split=0.2,
    subset="both",
    seed=123,
    image_size=[227,227],
    label_mode='int')
```

```
Found 262625 files belonging to 41 classes.
Using 210100 files for training.
Using 52525 files for validation.
```

[ ]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import␣
 ↪Conv2D,MaxPool2D,Dense,Flatten,Dropout,Input, AveragePooling2D,␣
 ↪Activation,Conv2D, MaxPooling2D, BatchNormalization,Concatenate
alexmodel = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(filters=96, kernel_size=(11,11), strides=(4,4),␣
 ↪activation='relu', input_shape=(227,227,3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Conv2D(filters=256, kernel_size=(5,5), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Conv2D(filters=256, kernel_size=(3,3), strides=(1,1),␣
 ↪activation='relu', padding="same"),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(4096, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(num_classes, activation='softmax')
])

# printing the model summary
alexmodel.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_10 (Conv2D)          (None, 55, 55, 96)        34944

 batch_normalization_10 (Bat  (None, 55, 55, 96)       384
 chNormalization)

 max_pooling2d_6 (MaxPooling  (None, 27, 27, 96)       0
 2D)

 conv2d_11 (Conv2D)          (None, 27, 27, 256)       614656
 batch_normalization_11 (Bat  (None, 27, 27, 256)      1024
 chNormalization)

 max_pooling2d_7 (MaxPooling  (None, 13, 13, 256)      0
 2D)

 conv2d_12 (Conv2D)          (None, 13, 13, 384)      885120

 batch_normalization_12 (Bat  (None, 13, 13, 384)      1536
 chNormalization)

 conv2d_13 (Conv2D)          (None, 13, 13, 384)      1327488

 batch_normalization_13 (Bat  (None, 13, 13, 384)      1536
 chNormalization)

 conv2d_14 (Conv2D)          (None, 13, 13, 256)      884992

 batch_normalization_14 (Bat  (None, 13, 13, 256)      1024
 chNormalization)

 max_pooling2d_8 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 flatten_2 (Flatten)         (None, 9216)             0

 dense_6 (Dense)             (None, 4096)             37752832

 dropout_4 (Dropout)         (None, 4096)             0

 dense_7 (Dense)             (None, 4096)             16781312

 dropout_5 (Dropout)         (None, 4096)             0

 dense_8 (Dense)             (None, 41)               167977

=================================================================
Total params: 58,454,825
Trainable params: 58,452,073
Non-trainable params: 2,752
_________________________________________________________________
```

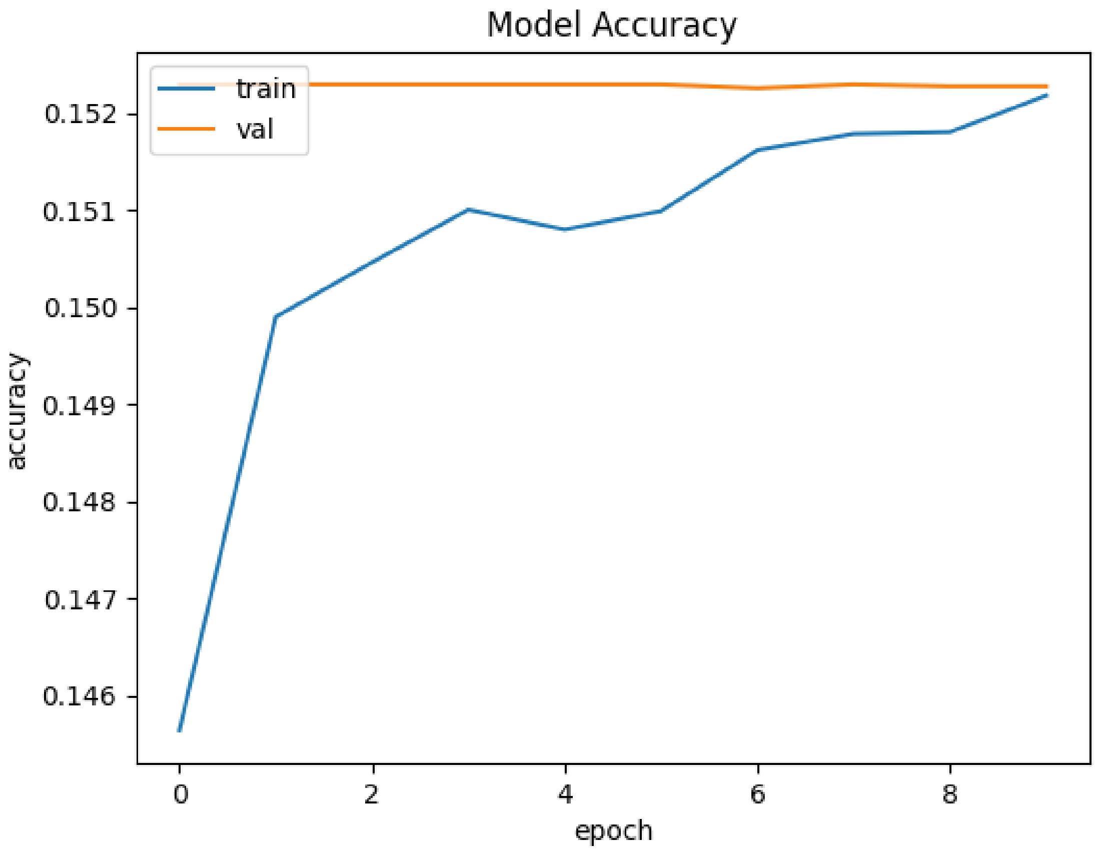
[ ]:

```
#Call back 1:
base_learning_rate = 0.001
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='accuracy', min_delta=0.005, patience=6, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

#call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1,callback_2]

#compiling our Model for datasert
alexmodel.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = alexmodel.fit(train_alexvmmrdb, epochs=30,␣
 ↪validation_data=test_alexvmmrdb,callbacks=callback_list)
```

```
Epoch 1/30
6566/6566 [==============================] - 121s 18ms/step - loss: 3.1978 -
accuracy: 0.1456 - val_loss: 3.0159 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 2/30
6566/6566 [==============================] - 116s 18ms/step - loss: 3.0235 -
accuracy: 0.1499 - val_loss: 3.0155 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 3/30
6566/6566 [==============================] - 115s 17ms/step - loss: 3.0223 -
accuracy: 0.1505 - val_loss: 3.0155 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 4/30
6566/6566 [==============================] - 116s 18ms/step - loss: 3.0214 -
accuracy: 0.1510 - val_loss: 3.0153 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 5/30
6566/6566 [==============================] - 115s 17ms/step - loss: 3.0563 -
accuracy: 0.1508 - val_loss: 3.0150 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 6/30
6566/6566 [==============================] - 114s 17ms/step - loss: 3.0275 -
accuracy: 0.1510 - val_loss: 3.0150 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 7/30
6566/6566 [==============================] - 116s 18ms/step - loss: 3.0209 -
accuracy: 0.1516 - val_loss: 3.0165 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 8/30
6566/6566 [==============================] - 117s 18ms/step - loss: 3.0208 -
accuracy: 0.1518 - val_loss: 3.0169 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 9/30
```

```
6566/6566 [==============================] - 116s 18ms/step - loss: 3.0202 -
accuracy: 0.1518 - val_loss: 3.0208 - val_accuracy: 0.1523 - lr: 0.0010
Epoch 10/30
6566/6566 [==============================] - 114s 17ms/step - loss: 3.0187 -
accuracy: 0.1522 - val_loss: 3.0198 - val_accuracy: 0.1523 - lr: 1.0000e-04
```

[ ]:
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



Up until now, only AlexNet has been used as an architecture, and the latest attempt had only 41 classes with over 200,000 images for training (approx. 5,000 images per class) but still results in an unacceptable level of accuracy.

In researching other papers that cited this data set, it seems to be common practice for researchers to only use a small portion of the data set and/or combine classes into 'make/model' or 'make only' classes. The most common way this data set is utilized is by identifying the top classes with a certain number of images, and ignoring all other classes.

As a result, the next steps will be to use only the top ______ classes with the most amount of images, and a few other architechtures will be explored.

1.7 Re-Analyzing VMMRdb for Classes With the Most Images

1.7.1 Extracting Original Data Set (again; performed in a new runtime)

[ ]:

```
import os
import shutil
from google.colab import drive

DIRPATH = "/content/VMMRdb/"

drive.mount('/content/drive/')
shutil.unpack_archive("/content/drive/MyDrive/endg511project/VMMRdb.zip", "/
  ↪content/")
print(len(os.listdir(DIRPATH)))
```

```
Mounted at /content/drive/
8174
```

1.7.2 VMMRdb Re-Analysis and Explanation of Next Steps

Reading Class Names and Number of Images into a Dataframe and sorting by Number of Images

[ ]:

```
import pandas as pd

df = pd.DataFrame(columns=["class_name", "num_images"])
for classname in os.listdir(DIRPATH):
  number_of_images = len(os.listdir(DIRPATH+"/"+classname))
  df = pd.concat([df, pd.DataFrame({"class_name": [classname], "num_images":␣
  ↪[number_of_images]})])

df = df.sort_values(by=["num_images"], ascending=False)
print(df)
```

```
              class_name num_images
0     ford_explorer_2002        883
0     nissan_altima_2005        716
0     ford_explorer_2003        584
0       honda_civic_2002        535
0     ford_explorer_2004        526
..                   …          …
0      nissan_200sx_1987          1
0       porsche_911_1974          1
0    pontiac_sunbird_2003          1
0     dodge_cummins_2000          1
```

```
0           bmw_528_2013          0

[8174 rows x 2 columns]
```

How many classes have 500 or more images?

[ ]:
```
filter_num = 500
print(df[df["num_images"] >= filter_num])
print(f'\nNumber of classes with >={filter_num} images: {df[df["num_images"] >␣
  ↪filter_num].shape[0]}')
```

```
           class_name num_images
0  ford_explorer_2002        883
0  nissan_altima_2005        716
0  ford_explorer_2003        584
0    honda_civic_2002        535
0  ford_explorer_2004        526
0   ford_mustang_2000        504

Number of classes with >=500 images: 6
```

This is probably too few classes. Try 400 or more classes.

[ ]:
```
filter_num = 400
print(df[df["num_images"] >= filter_num])
print(f'\nNumber of classes with >={filter_num} images: {df[df["num_images"] >␣
  ↪filter_num].shape[0]}')
```

```
                 class_name num_images
0        ford_explorer_2002        883
0        nissan_altima_2005        716
0        ford_explorer_2003        584
0          honda_civic_2002        535
0        ford_explorer_2004        526
0         ford_mustang_2000        504
0  chevrolet_silverado_2004        487
0         toyota_camry_2007        467
0        nissan_altima_2002        458
0  dodge_grand caravan_2005        447
0          honda_civic_1998        443
0         honda_accord_2000        440
0        nissan_altima_2003        438
0         honda_accord_1999        437
0     chevrolet_impala_2006        436
0          honda_civic_2001        434
0         honda_accord_1998        429
0         honda_accord_2003        420
0        nissan_altima_2006        419
0            ford_f150_2004        414
```

```
0          ford_taurus_2001        410
0         honda_accord_2005        409
0        nissan_altima_2008        408
0          ford_taurus_2003        405
0     volkswagen_jetta_2006        402

Number of classes with >=400 images: 25
```

Note that there are some classes with the same make/model but only differ by one year. This may be difficult for a ML model to discern between. If these make/model classes are combined, how many classes will be left?

```
[ ]: filtered_df = df[df["num_images"] >= filter_num].copy()
     filtered_df.loc[:, "make_model"] = filtered_df["class_name"].str.split("_").
      ↪str[:2].str.join("_")
     df_make_model = filtered_df[["make_model", "num_images"]].groupby("make_model").
      ↪sum()
     print(df_make_model)
     print(f'\nNumber of make/model classes after combining classes with␣
      ↪>={filter_num} images: {df_make_model.shape[0]}')
```

```
                     num_images
make_model
chevrolet_impala            436
chevrolet_silverado         487
dodge_grand caravan         447
ford_explorer              1993
ford_f150                   414
ford_mustang                504
ford_taurus                 815
honda_accord               2135
honda_civic                1412
nissan_altima              2439
toyota_camry                467
volkswagen_jetta            402

Number of make/model classes after combining classes with >=400 images: 12
```

We will initially try using only classes that have >= 400 images, and combine them into make/model classes if performance is low.

Given that the use case is a mobile processing unit of dashcam footage, MobileNetV2 and MobileNetV3 (`MobileNetV3Large`) will be explored as the next architectures.

1.7.3 Create Filtered Data Set

```
[ ]: NEW_DIR = '/content/VMMRdb_LrgImgCount/'
     os.mkdir(NEW_DIR)
```

[ ]:

```
# Loop through each folder in the directory
for folder in os.listdir(DIRPATH):
    # Check if the folder name exists in the specified column of the dataframe
    if folder in filtered_df["class_name"].values:
        # Create a new directory in the destination directory with the same␣
 ↪name as the matching folder
        new_folder_path = os.path.join(NEW_DIR, folder)
        os.makedirs(new_folder_path, exist_ok=True)
        # Copy the images in the matching folder to the new directory
        src_folder_path = os.path.join(DIRPATH, folder)
        for file_name in os.listdir(src_folder_path):
            src_file_path = os.path.join(src_folder_path, file_name)
            dest_file_path = os.path.join(new_folder_path, file_name)
            shutil.copyfile(src_file_path, dest_file_path)
```

[ ]:

```
print(f'rows of filtered dataframe: {filtered_df.shape[0]}')
print(f'number of classes in new directory: {len(os.listdir(NEW_DIR))}')
```

```
rows of filtered dataframe: 25
number of classes in new directory: 25
```

1.7.4 Zip Filtered Dataset for later use.

[ ]:

```
zip_filename = 'VMMRdb_LrgImgCount'
shutil.make_archive(os.path.join('/content/drive/MyDrive/endg511project/
 ↪',zip_filename), 'zip', NEW_DIR)
```

[ ]:

```
'/content/drive/MyDrive/endg511project/VMMRdb_LrgImgCount.zip'
```

1.8 Creating a Save Directory for Models

[ ]:

```
SAVE_DIR = '/content/drive/MyDrive/endg511project/SavedModels/'
```

[ ]:

```
os.mkdir(SAVE_DIR)
```

1.9 Train MobileNetV2 with Filtered Datset

[ ]:

```
import os
import shutil
from google.colab import drive
import tensorflow as tf
from matplotlib import pyplot as plt

DIRPATH = '/content/VMMRdb_LrgImgCount/'
SAVE_DIR = '/content/drive/MyDrive/endg511project/SavedModels/'

drive.mount('/content/drive/')
```

```
shutil.unpack_archive("/content/drive/MyDrive/endg511project/VMMRdb_LrgImgCount.
 ↪zip", DIRPATH)

num_classes = len(os.listdir(DIRPATH))
print(f"Number of Classes: {num_classes}")
for subdir in os.listdir(DIRPATH):
  print(f"Class: {subdir}, Count: {len(os.listdir(os.path.join(DIRPATH,␣
 ↪subdir)))}")
```

```
Mounted at /content/drive/
Number of Classes: 25
Class: ford_explorer_2003, Count: 584
Class: honda_accord_2000, Count: 440
Class: ford_f150_2004, Count: 414
Class: honda_accord_1999, Count: 437
Class: ford_taurus_2001, Count: 410
Class: honda_accord_1998, Count: 429
Class: nissan_altima_2005, Count: 716
Class: ford_mustang_2000, Count: 504
Class: nissan_altima_2008, Count: 408
Class: ford_explorer_2002, Count: 883
Class: chevrolet_silverado_2004, Count: 487
Class: ford_explorer_2004, Count: 526
Class: nissan_altima_2003, Count: 438
Class: nissan_altima_2006, Count: 419
Class: volkswagen_jetta_2006, Count: 402
Class: honda_accord_2005, Count: 409
Class: honda_accord_2003, Count: 420
Class: honda_civic_2002, Count: 535
Class: chevrolet_impala_2006, Count: 436
Class: honda_civic_2001, Count: 434
Class: honda_civic_1998, Count: 443
Class: nissan_altima_2002, Count: 458
Class: toyota_camry_2007, Count: 467
Class: dodge_grand caravan_2005, Count: 447
Class: ford_taurus_2003, Count: 405
```

```
[ ]: train_mobilenet, test_mobilenet = tf.keras.utils.image_dataset_from_directory(
         DIRPATH,
         validation_split=0.2,
         subset="both",
         seed=123,
         image_size=[224,224],
         label_mode='int')
```

```
Found 11951 files belonging to 25 classes.
Using 9561 files for training.
Using 2390 files for validation.
```

Computing Class Weights

[ ]:

```python
import numpy as np
from collections import Counter

# Get the class labels from the dataset
class_labels = train_mobilenet.class_names

# Get the number of images per class
num_images_per_class = []
for images, labels in train_mobilenet:
    for label in labels.numpy():
        num_images_per_class.append(label)
class_counts = Counter(num_images_per_class)

# Calculate class frequencies
total_num_images = sum(class_counts.values())
class_frequencies = []
for i in range(len(class_labels)):
    class_frequencies.append(class_counts[i] / total_num_images)

# Calculate class weights
max_frequency = max(class_frequencies)
class_weights = {}
for i in range(len(class_labels)):
    class_weights[i] = max_frequency / class_frequencies[i]

print(class_weights)
```

```
{0: 1.9744318181818186, 1: 1.7506297229219145, 2: 1.9146005509641875, 3: 1.0, 4:
1.4509394572025052, 5: 1.6951219512195124, 6: 2.068452380952381, 7:
1.759493670886076, 8: 2.1517027863777094, 9: 2.131901840490798, 10:
2.0028818443804037, 11: 1.9198895027624312, 12: 1.9522471910112362, 13:
2.0746268656716422, 14: 2.0746268656716422, 15: 1.9744318181818186, 16:
2.0562130177514795, 17: 1.6162790697674418, 18: 1.8733153638814017, 19:
1.9632768361581923, 20: 1.2477558348294435, 21: 2.165109034267913, 22:
2.131901840490798, 23: 1.8733153638814017, 24: 2.1060606060606064}
```

Creating the model

[ ]:

```python
mobilev2model = tf.keras.applications.MobileNetV2(
    input_shape=(224,224,3),
    alpha=1.0,
    include_top=True,
    weights=None,
    input_tensor=None,
    pooling='max',
    classes=num_classes
)
mobilev2model.summary()
```

```
Model: "mobilenetv2_1.00_224"
--------------------------------------------------------------------------------
------------------
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864
['input_1[0][0]']
                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0][0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0
['bn_Conv1[0][0]']
                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288
['Conv1_relu[0][0]']
 wiseConv2D)                    )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128
['expanded_conv_depthwise[0][0]']
 tchNormalization)              )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0
['expanded_conv_depthwise_BN[0][0
 ReLU)                          )                                ]']

 expanded_conv_project (Conv2D)  (None, 112, 112, 16  512
['expanded_conv_depthwise_relu[0]
                                )                                [0]']

 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64
['expanded_conv_project[0][0]']
 hNormalization)                )

 block_1_expand (Conv2D)        (None, 112, 112, 96  1536
['expanded_conv_project_BN[0][0]'
                                )                                ]
 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384
['block_1_expand[0][0]']
 ization)                       )

 block_1_expand_relu (ReLU)     (None, 112, 112, 96  0
['block_1_expand_BN[0][0]']
                                )

 block_1_pad (ZeroPadding2D)    (None, 113, 113, 96  0
['block_1_expand_relu[0][0]']
                                )

 block_1_depthwise (DepthwiseCo  (None, 56, 56, 96)   864
['block_1_pad[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 56, 56, 96)   384
['block_1_depthwise[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 56, 56, 96)   0
['block_1_depthwise_BN[0][0]']

 block_1_project (Conv2D)       (None, 56, 56, 24)   2304
['block_1_depthwise_relu[0][0]']

 block_1_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_1_project[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_1_project_BN[0][0]']

 block_2_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_2_expand[0][0]']
 ization)

 block_2_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_2_expand_BN[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 56, 56, 144)  1296
['block_2_expand_relu[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 56, 56, 144)  576
['block_2_depthwise[0][0]']
 malization)
 block_2_depthwise_relu (ReLU)  (None, 56, 56, 144)  0
['block_2_depthwise_BN[0][0]']

 block_2_project (Conv2D)       (None, 56, 56, 24)   3456
['block_2_depthwise_relu[0][0]']

 block_2_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_2_project[0][0]']
 lization)

 block_2_add (Add)              (None, 56, 56, 24)   0
['block_1_project_BN[0][0]',
'block_2_project_BN[0][0]']

 block_3_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_2_add[0][0]']

 block_3_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_3_expand[0][0]']
 ization)

 block_3_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_3_expand_BN[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 57, 57, 144)  0
['block_3_expand_relu[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 28, 28, 144)  1296
['block_3_pad[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 28, 28, 144)  576
['block_3_depthwise[0][0]']
 malization)

 block_3_depthwise_relu (ReLU)  (None, 28, 28, 144)  0
['block_3_depthwise_BN[0][0]']

 block_3_project (Conv2D)       (None, 28, 28, 32)   4608
['block_3_depthwise_relu[0][0]']

 block_3_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_3_project[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_3_project_BN[0][0]']
```

```
 block_4_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_4_expand[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_4_expand_BN[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_4_expand_relu[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_4_depthwise[0][0]']
 malization)

 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_4_depthwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 28, 28, 32)   6144
['block_4_depthwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_4_project[0][0]']
 lization)

 block_4_add (Add)              (None, 28, 28, 32)   0
['block_3_project_BN[0][0]',
'block_4_project_BN[0][0]']

 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_4_add[0][0]']

 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_5_expand[0][0]']
 ization)

 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_5_expand_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_5_expand_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_5_depthwise[0][0]']
 malization)
 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_5_depthwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 28, 28, 32)   6144
['block_5_depthwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_5_project[0][0]']
 lization)

 block_5_add (Add)              (None, 28, 28, 32)   0
['block_4_add[0][0]',
'block_5_project_BN[0][0]']

 block_6_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_5_add[0][0]']

 block_6_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_6_expand[0][0]']
 ization)

 block_6_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_6_expand_BN[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 29, 29, 192)  0
['block_6_expand_relu[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 14, 14, 192)  1728
['block_6_pad[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 14, 14, 192)  768
['block_6_depthwise[0][0]']
 malization)

 block_6_depthwise_relu (ReLU)  (None, 14, 14, 192)  0
['block_6_depthwise_BN[0][0]']

 block_6_project (Conv2D)       (None, 14, 14, 64)   12288
['block_6_depthwise_relu[0][0]']

 block_6_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_6_project[0][0]']
 lization)

 block_7_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_6_project_BN[0][0]']
```

```
 block_7_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_7_expand[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_7_expand_BN[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_7_expand_relu[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_7_depthwise[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_7_depthwise_BN[0][0]']

 block_7_project (Conv2D)       (None, 14, 14, 64)   24576
['block_7_depthwise_relu[0][0]']

 block_7_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_7_project[0][0]']
 lization)

 block_7_add (Add)              (None, 14, 14, 64)   0
['block_6_project_BN[0][0]',
'block_7_project_BN[0][0]']

 block_8_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_7_add[0][0]']

 block_8_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_8_expand[0][0]']
 ization)

 block_8_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_8_expand_BN[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_8_expand_relu[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_8_depthwise[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
```

```
['block_8_depthwise_BN[0][0]']

 block_8_project (Conv2D)       (None, 14, 14, 64)   24576
['block_8_depthwise_relu[0][0]']

 block_8_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_8_project[0][0]']
 lization)

 block_8_add (Add)              (None, 14, 14, 64)   0
['block_7_add[0][0]',
'block_8_project_BN[0][0]']

 block_9_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_8_add[0][0]']

 block_9_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_9_expand[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_9_expand_BN[0][0]']

 block_9_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_9_expand_relu[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_9_depthwise[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_9_depthwise_BN[0][0]']

 block_9_project (Conv2D)       (None, 14, 14, 64)   24576
['block_9_depthwise_relu[0][0]']

 block_9_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_9_project[0][0]']
 lization)

 block_9_add (Add)              (None, 14, 14, 64)   0
['block_8_add[0][0]',
'block_9_project_BN[0][0]']

 block_10_expand (Conv2D)       (None, 14, 14, 384)  24576
['block_9_add[0][0]']
```

```
 block_10_expand_BN (BatchNorma  (None, 14, 14, 384)  1536
['block_10_expand[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 14, 14, 384)  0
['block_10_expand_BN[0][0]']

 block_10_depthwise (DepthwiseC  (None, 14, 14, 384)  3456
['block_10_expand_relu[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 14, 14, 384)  1536
['block_10_depthwise[0][0]']
 rmalization)

 block_10_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_10_depthwise_BN[0][0]']

 block_10_project (Conv2D)      (None, 14, 14, 96)   36864
['block_10_depthwise_relu[0][0]']

 block_10_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_10_project[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_10_project_BN[0][0]']

 block_11_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_11_expand[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_11_expand_BN[0][0]']

 block_11_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_11_expand_relu[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_11_depthwise[0][0]']
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
['block_11_depthwise_BN[0][0]']

 block_11_project (Conv2D)      (None, 14, 14, 96)   55296
['block_11_depthwise_relu[0][0]']
```

```
 block_11_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_11_project[0][0]']
 alization)

 block_11_add (Add)             (None, 14, 14, 96)   0
['block_10_project_BN[0][0]',
'block_11_project_BN[0][0]']

 block_12_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_11_add[0][0]']

 block_12_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_12_expand[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_12_expand_BN[0][0]']

 block_12_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_12_expand_relu[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_12_depthwise[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
['block_12_depthwise_BN[0][0]']

 block_12_project (Conv2D)      (None, 14, 14, 96)   55296
['block_12_depthwise_relu[0][0]']

 block_12_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_12_project[0][0]']
 alization)

 block_12_add (Add)             (None, 14, 14, 96)   0
['block_11_add[0][0]',
'block_12_project_BN[0][0]']

 block_13_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_12_add[0][0]']

 block_13_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_13_expand[0][0]']
 lization)
 block_13_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_13_expand_BN[0][0]']

 block_13_pad (ZeroPadding2D)   (None, 15, 15, 576)  0
['block_13_expand_relu[0][0]']

 block_13_depthwise (DepthwiseC  (None, 7, 7, 576)   5184
['block_13_pad[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304
['block_13_depthwise[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 7, 7, 576)   0
['block_13_depthwise_BN[0][0]']

 block_13_project (Conv2D)      (None, 7, 7, 160)    92160
['block_13_depthwise_relu[0][0]']

 block_13_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_13_project[0][0]']
 alization)

 block_14_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_13_project_BN[0][0]']

 block_14_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_14_expand[0][0]']
 lization)

 block_14_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_14_expand_BN[0][0]']

 block_14_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_14_expand_relu[0][0]']
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_14_depthwise[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_14_depthwise_BN[0][0]']

 block_14_project (Conv2D)      (None, 7, 7, 160)    153600
['block_14_depthwise_relu[0][0]']
```

```
 block_14_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_14_project[0][0]']
 alization)

 block_14_add (Add)             (None, 7, 7, 160)    0
['block_13_project_BN[0][0]',
'block_14_project_BN[0][0]']

 block_15_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_14_add[0][0]']

 block_15_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_15_expand[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_15_expand_BN[0][0]']

 block_15_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_15_expand_relu[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_15_depthwise[0][0]']
 rmalization)

 block_15_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_15_depthwise_BN[0][0]']

 block_15_project (Conv2D)      (None, 7, 7, 160)    153600
['block_15_depthwise_relu[0][0]']

 block_15_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_15_project[0][0]']
 alization)

 block_15_add (Add)             (None, 7, 7, 160)    0
['block_14_add[0][0]',
'block_15_project_BN[0][0]']

 block_16_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_15_add[0][0]']

 block_16_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_16_expand[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 7, 7, 960)    0
```

```
['block_16_expand_BN[0][0]']

 block_16_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_16_expand_relu[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_16_depthwise[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_16_depthwise_BN[0][0]']

 block_16_project (Conv2D)      (None, 7, 7, 320)    307200
['block_16_depthwise_relu[0][0]']

 block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280
['block_16_project[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600
['block_16_project_BN[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120
['Conv_1[0][0]']

 out_relu (ReLU)                (None, 7, 7, 1280)   0
['Conv_1_bn[0][0]']

 global_average_pooling2d (Glob  (None, 1280)        0
['out_relu[0][0]']
 alAveragePooling2D)

 predictions (Dense)            (None, 25)           32025
['global_average_pooling2d[0][0]'
                                                                 ]

================================================================================
==================
Total params: 2,290,009
Trainable params: 2,255,897
Non-trainable params: 34,112
--------------------------------------------------------------------------------
------------------
```

```
[ ]: # Call back 1:
     base_learning_rate = 1e-4
```

```python
opt1 = tf.keras.optimizers.Adam(learning_rate=base_learning_rate)
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='accuracy', min_delta=0, patience=10, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

# Call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1, callback_2]

#compiling our Model for dataset
mobilev2model.compile(optimizer=opt1,
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = mobilev2model.fit(
    train_mobilenet,
    epochs=60,
    validation_data=test_mobilenet,
    class_weight=class_weights,
    callbacks=callback_list)
```

```
Epoch 1/60
299/299 [==============================] - 42s 51ms/step - loss: 1.3453 -
accuracy: 0.7565 - val_loss: 4.7913 - val_accuracy: 0.1515 - lr: 1.0000e-04
Epoch 2/60
299/299 [==============================] - 15s 49ms/step - loss: 1.2569 -
accuracy: 0.7710 - val_loss: 4.8691 - val_accuracy: 0.1410 - lr: 1.0000e-04
Epoch 3/60
299/299 [==============================] - 15s 49ms/step - loss: 1.0572 -
accuracy: 0.8036 - val_loss: 5.7748 - val_accuracy: 0.1423 - lr: 1.0000e-04
Epoch 4/60
299/299 [==============================] - 15s 49ms/step - loss: 0.8951 -
accuracy: 0.8420 - val_loss: 6.1853 - val_accuracy: 0.1393 - lr: 1.0000e-04
Epoch 5/60
299/299 [==============================] - 15s 49ms/step - loss: 0.7976 -
accuracy: 0.8563 - val_loss: 5.8315 - val_accuracy: 0.1406 - lr: 1.0000e-04
Epoch 6/60
299/299 [==============================] - 15s 49ms/step - loss: 0.5434 -
accuracy: 0.9074 - val_loss: 4.4600 - val_accuracy: 0.1749 - lr: 1.0000e-05
```

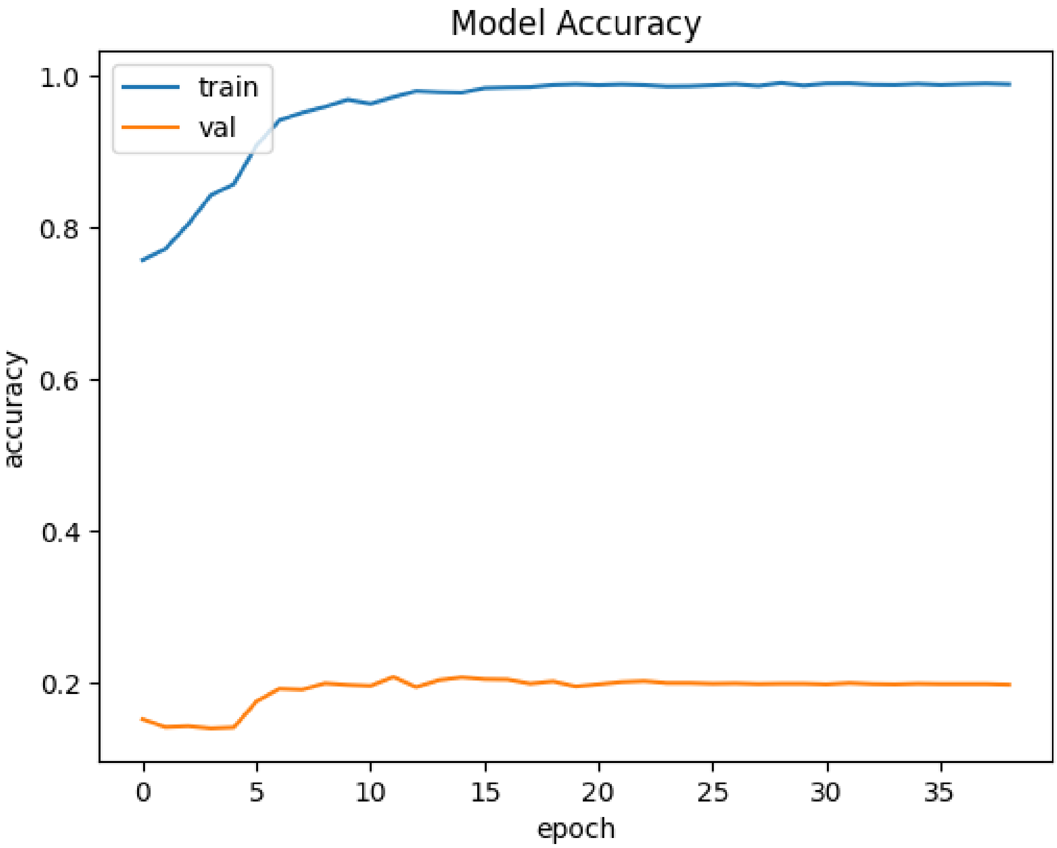
```
Epoch 7/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3897 -
accuracy: 0.9406 - val_loss: 4.1217 - val_accuracy: 0.1916 - lr: 1.0000e-05
Epoch 8/60
299/299 [==============================] - 15s 48ms/step - loss: 0.3408 -
accuracy: 0.9505 - val_loss: 4.0577 - val_accuracy: 0.1904 - lr: 1.0000e-05
Epoch 9/60
299/299 [==============================] - 15s 49ms/step - loss: 0.2989 -
accuracy: 0.9584 - val_loss: 3.9824 - val_accuracy: 0.1987 - lr: 1.0000e-05
Epoch 10/60
299/299 [==============================] - 15s 49ms/step - loss: 0.2618 -
accuracy: 0.9678 - val_loss: 3.9681 - val_accuracy: 0.1967 - lr: 1.0000e-05
Epoch 11/60
299/299 [==============================] - 15s 48ms/step - loss: 0.2669 -
accuracy: 0.9625 - val_loss: 3.9529 - val_accuracy: 0.1954 - lr: 1.0000e-05
Epoch 12/60
299/299 [==============================] - 15s 49ms/step - loss: 0.2312 -
accuracy: 0.9711 - val_loss: 3.9764 - val_accuracy: 0.2071 - lr: 1.0000e-05
Epoch 13/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1976 -
accuracy: 0.9790 - val_loss: 4.0063 - val_accuracy: 0.1937 - lr: 1.0000e-05
Epoch 14/60
299/299 [==============================] - 15s 48ms/step - loss: 0.2009 -
accuracy: 0.9776 - val_loss: 3.9284 - val_accuracy: 0.2029 - lr: 1.0000e-05
Epoch 15/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1884 -
accuracy: 0.9770 - val_loss: 3.9524 - val_accuracy: 0.2067 - lr: 1.0000e-05
Epoch 16/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1704 -
accuracy: 0.9831 - val_loss: 3.9848 - val_accuracy: 0.2042 - lr: 1.0000e-05
Epoch 17/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1545 -
accuracy: 0.9839 - val_loss: 3.9786 - val_accuracy: 0.2038 - lr: 1.0000e-05
Epoch 18/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1459 -
accuracy: 0.9843 - val_loss: 4.0445 - val_accuracy: 0.1983 - lr: 1.0000e-05
Epoch 19/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1336 -
accuracy: 0.9871 - val_loss: 4.0174 - val_accuracy: 0.2013 - lr: 1.0000e-06
Epoch 20/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1339 -
accuracy: 0.9881 - val_loss: 4.0035 - val_accuracy: 0.1946 - lr: 1.0000e-06
Epoch 21/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1269 -
accuracy: 0.9869 - val_loss: 4.0027 - val_accuracy: 0.1975 - lr: 1.0000e-06
Epoch 22/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1288 -
accuracy: 0.9879 - val_loss: 4.0081 - val_accuracy: 0.2004 - lr: 1.0000e-06
```

```
Epoch 23/60
299/299 [==============================] - 14s 48ms/step - loss: 0.1225 -
accuracy: 0.9870 - val_loss: 4.0076 - val_accuracy: 0.2017 - lr: 1.0000e-07
Epoch 24/60
299/299 [==============================] - 14s 48ms/step - loss: 0.1428 -
accuracy: 0.9851 - val_loss: 4.0068 - val_accuracy: 0.1992 - lr: 1.0000e-07
Epoch 25/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1356 -
accuracy: 0.9855 - val_loss: 4.0074 - val_accuracy: 0.1992 - lr: 1.0000e-07
Epoch 26/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1365 -
accuracy: 0.9868 - val_loss: 4.0071 - val_accuracy: 0.1983 - lr: 1.0000e-07
Epoch 27/60
299/299 [==============================] - 14s 48ms/step - loss: 0.1261 -
accuracy: 0.9882 - val_loss: 4.0072 - val_accuracy: 0.1987 - lr: 1.0000e-08
Epoch 28/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1306 -
accuracy: 0.9859 - val_loss: 4.0070 - val_accuracy: 0.1979 - lr: 1.0000e-08
Epoch 29/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1208 -
accuracy: 0.9898 - val_loss: 4.0077 - val_accuracy: 0.1983 - lr: 1.0000e-08
Epoch 30/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1326 -
accuracy: 0.9863 - val_loss: 4.0078 - val_accuracy: 0.1983 - lr: 1.0000e-08
Epoch 31/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1221 -
accuracy: 0.9890 - val_loss: 4.0078 - val_accuracy: 0.1975 - lr: 1.0000e-09
Epoch 32/60
299/299 [==============================] - 14s 48ms/step - loss: 0.1229 -
accuracy: 0.9892 - val_loss: 4.0086 - val_accuracy: 0.1992 - lr: 1.0000e-09
Epoch 33/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1274 -
accuracy: 0.9874 - val_loss: 4.0085 - val_accuracy: 0.1979 - lr: 1.0000e-09
Epoch 34/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1271 -
accuracy: 0.9871 - val_loss: 4.0077 - val_accuracy: 0.1975 - lr: 1.0000e-09
Epoch 35/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1246 -
accuracy: 0.9885 - val_loss: 4.0078 - val_accuracy: 0.1983 - lr: 1.0000e-10
Epoch 36/60
299/299 [==============================] - 15s 48ms/step - loss: 0.1346 -
accuracy: 0.9871 - val_loss: 4.0080 - val_accuracy: 0.1979 - lr: 1.0000e-10
Epoch 37/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1215 -
accuracy: 0.9882 - val_loss: 4.0081 - val_accuracy: 0.1979 - lr: 1.0000e-10
Epoch 38/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1208 -
accuracy: 0.9889 - val_loss: 4.0083 - val_accuracy: 0.1979 - lr: 1.0000e-10
```

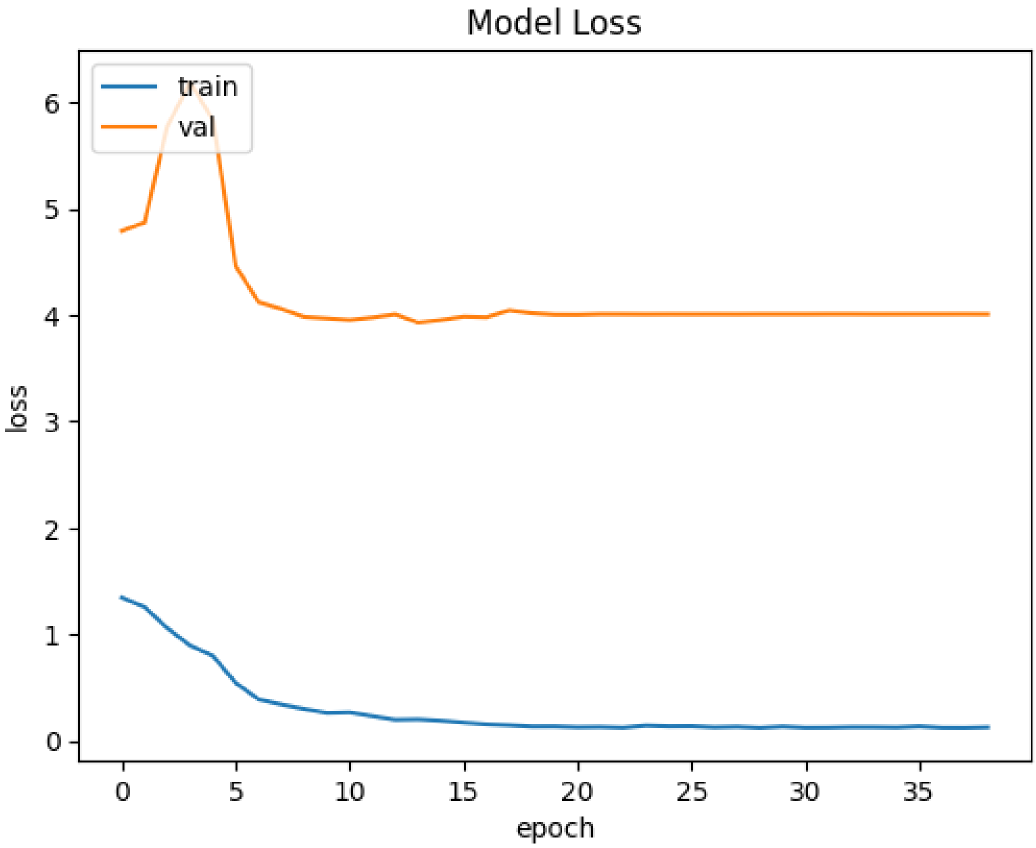
```
Epoch 39/60
299/299 [==============================] - 15s 49ms/step - loss: 0.1256 -
accuracy: 0.9880 - val_loss: 4.0074 - val_accuracy: 0.1971 - lr: 1.0000e-11
```

Training was ran three consecutive times, and stopped early on the final two runs; approximately 150 epochs in total. This is the highest validation accuracy that could be achieved.

```
[ ]: plt.plot(history.history['accuracy'])
     plt.plot(history.history['val_accuracy'])
     plt.title('Model Accuracy')
     plt.ylabel('accuracy')
     plt.xlabel('epoch')
     plt.legend(['train', 'val'], loc='upper left')
     plt.show()
```



```
[ ]: plt.plot(history.history['loss'])
     plt.plot(history.history['val_loss'])
     plt.title('Model Loss')
     plt.ylabel('loss')
     plt.xlabel('epoch')
     plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



[ ]:
```
mobilev2model.save(os.path.join(SAVE_DIR,'mobileNetV2_noPretrain_noTuning.h5'))
```

1.10 Train MobileNetV3 with Filtered Dataset

[ ]:
```
mobilev3model = tf.keras.applications.MobileNetV3Large(
    input_shape=(224,224,3),
    alpha=1.0,
    include_top=True,
    weights=None,
    pooling='max'
)
mobilev3model.summary()
```

```
Model: "MobilenetV3large"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_2 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 rescaling (Rescaling)          (None, 224, 224, 3)  0
['input_2[0][0]']

 Conv (Conv2D)                  (None, 112, 112, 16  432
['rescaling[0][0]']
                                )

 Conv/BatchNorm (BatchNormaliza  (None, 112, 112, 16  64          ['Conv[0][0]']
 tion)                          )

 tf.__operators__.add (TFOpLamb  (None, 112, 112, 16  0
['Conv/BatchNorm[0][0]']
 da)                            )

 re_lu (ReLU)                   (None, 112, 112, 16  0
['tf.__operators__.add[0][0]']
                                )

 tf.math.multiply (TFOpLambda)  (None, 112, 112, 16  0           ['re_lu[0][0]']
                                )

 multiply (Multiply)            (None, 112, 112, 16  0
['Conv/BatchNorm[0][0]',
                                )
'tf.math.multiply[0][0]']

 expanded_conv/depthwise (Depth  (None, 112, 112, 16  144
['multiply[0][0]']
 wiseConv2D)                    )

 expanded_conv/depthwise/BatchN  (None, 112, 112, 16  64
['expanded_conv/depthwise[0][0]']
 orm (BatchNormalization)       )

 re_lu_1 (ReLU)                 (None, 112, 112, 16  0
['expanded_conv/depthwise/BatchNo
                                )                                rm[0][0]']

 expanded_conv/project (Conv2D)  (None, 112, 112, 16  256
['re_lu_1[0][0]']
                                )

 expanded_conv/project/BatchNor  (None, 112, 112, 16  64
['expanded_conv/project[0][0]']
```

```
 m (BatchNormalization)         )

 expanded_conv/Add (Add)        (None, 112, 112, 16  0
['multiply[0][0]',
                                )
'expanded_conv/project/BatchNorm
                                                                 [0][0]']

 expanded_conv_1/expand (Conv2D  (None, 112, 112, 64  1024
['expanded_conv/Add[0][0]']
 )                              )

 expanded_conv_1/expand/BatchNo  (None, 112, 112, 64  256
['expanded_conv_1/expand[0][0]']
 rm (BatchNormalization)        )

 re_lu_2 (ReLU)                 (None, 112, 112, 64  0
['expanded_conv_1/expand/BatchNor
                                )                                m[0][0]']

 expanded_conv_1/depthwise/pad   (None, 113, 113, 64  0
['re_lu_2[0][0]']
 (ZeroPadding2D)                )

 expanded_conv_1/depthwise (Dep  (None, 56, 56, 64)  576
['expanded_conv_1/depthwise/pad[0
 thwiseConv2D)                                                   ][0]']

 expanded_conv_1/depthwise/Batc  (None, 56, 56, 64)  256
['expanded_conv_1/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_3 (ReLU)                 (None, 56, 56, 64)   0
['expanded_conv_1/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_1/project (Conv2  (None, 56, 56, 24)  1536
['re_lu_3[0][0]']
 D)

 expanded_conv_1/project/BatchN  (None, 56, 56, 24)  96
['expanded_conv_1/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_2/expand (Conv2D  (None, 56, 56, 72)  1728
['expanded_conv_1/project/BatchNo
 )                                                               rm[0][0]']
 expanded_conv_2/expand/BatchNo  (None, 56, 56, 72)  288
['expanded_conv_2/expand[0][0]']
 rm (BatchNormalization)

 re_lu_4 (ReLU)                 (None, 56, 56, 72)   0
['expanded_conv_2/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_2/depthwise (Dep  (None, 56, 56, 72)  648
['re_lu_4[0][0]']
 thwiseConv2D)

 expanded_conv_2/depthwise/Batc  (None, 56, 56, 72)  288
['expanded_conv_2/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_5 (ReLU)                 (None, 56, 56, 72)   0
['expanded_conv_2/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_2/project (Conv2  (None, 56, 56, 24)  1728
['re_lu_5[0][0]']
 D)

 expanded_conv_2/project/BatchN  (None, 56, 56, 24)  96
['expanded_conv_2/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_2/Add (Add)      (None, 56, 56, 24)   0
['expanded_conv_1/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_2/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_3/expand (Conv2D  (None, 56, 56, 72)  1728
['expanded_conv_2/Add[0][0]']
 )

 expanded_conv_3/expand/BatchNo  (None, 56, 56, 72)  288
['expanded_conv_3/expand[0][0]']
 rm (BatchNormalization)

 re_lu_6 (ReLU)                 (None, 56, 56, 72)   0
['expanded_conv_3/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_3/depthwise/pad   (None, 59, 59, 72)  0
['re_lu_6[0][0]']
```

```
 (ZeroPadding2D)

 expanded_conv_3/depthwise (Dep  (None, 28, 28, 72)  1800
['expanded_conv_3/depthwise/pad[0
 thwiseConv2D)                                                   ][0]']

 expanded_conv_3/depthwise/Batc  (None, 28, 28, 72)  288
['expanded_conv_3/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_7 (ReLU)                 (None, 28, 28, 72)   0
['expanded_conv_3/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 72)    0
['re_lu_7[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_3/squeeze_excite  (None, 1, 1, 24)    1752
['expanded_conv_3/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 24)    0
['expanded_conv_3/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 72)    1800
['expanded_conv_3/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']

 tf.__operators__.add_1 (TFOpLa  (None, 1, 1, 72)    0
['expanded_conv_3/squeeze_excite/
 mbda)                                                           Conv_1[0][0]']

 re_lu_8 (ReLU)                 (None, 1, 1, 72)     0
['tf.__operators__.add_1[0][0]']

 tf.math.multiply_1 (TFOpLambda  (None, 1, 1, 72)    0
['re_lu_8[0][0]']
 )

 expanded_conv_3/squeeze_excite  (None, 28, 28, 72)  0
['re_lu_7[0][0]',
 /Mul (Multiply)
'tf.math.multiply_1[0][0]']

 expanded_conv_3/project (Conv2  (None, 28, 28, 40)  2880
['expanded_conv_3/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_3/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_3/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_4/expand (Conv2D  (None, 28, 28, 120)  4800
['expanded_conv_3/project/BatchNo
 )                                                               rm[0][0]']

 expanded_conv_4/expand/BatchNo  (None, 28, 28, 120)  480
['expanded_conv_4/expand[0][0]']
 rm (BatchNormalization)

 re_lu_9 (ReLU)                 (None, 28, 28, 120)  0
['expanded_conv_4/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_4/depthwise (Dep  (None, 28, 28, 120)  3000
['re_lu_9[0][0]']
 thwiseConv2D)

 expanded_conv_4/depthwise/Batc  (None, 28, 28, 120)  480
['expanded_conv_4/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_10 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_4/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 120)   0
['re_lu_10[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_4/squeeze_excite  (None, 1, 1, 32)    3872
['expanded_conv_4/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 32)    0
['expanded_conv_4/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 120)   3960
['expanded_conv_4/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']
 tf.__operators__.add_2 (TFOpLa  (None, 1, 1, 120)   0
['expanded_conv_4/squeeze_excite/
 mbda)                                                            Conv_1[0][0]']

 re_lu_11 (ReLU)                (None, 1, 1, 120)    0
['tf.__operators__.add_2[0][0]']

 tf.math.multiply_2 (TFOpLambda  (None, 1, 1, 120)   0
['re_lu_11[0][0]']
 )

 expanded_conv_4/squeeze_excite  (None, 28, 28, 120)  0
['re_lu_10[0][0]',
 /Mul (Multiply)
'tf.math.multiply_2[0][0]']

 expanded_conv_4/project (Conv2  (None, 28, 28, 40)  4800
['expanded_conv_4/squeeze_excite/
 D)                                                               Mul[0][0]']

 expanded_conv_4/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_4/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_4/Add (Add)      (None, 28, 28, 40)   0
['expanded_conv_3/project/BatchNo
                                                                  rm[0][0]',
'expanded_conv_4/project/BatchNo
                                                                  rm[0][0]']

 expanded_conv_5/expand (Conv2D  (None, 28, 28, 120)  4800
['expanded_conv_4/Add[0][0]']
 )

 expanded_conv_5/expand/BatchNo  (None, 28, 28, 120)  480
['expanded_conv_5/expand[0][0]']
 rm (BatchNormalization)

 re_lu_12 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_5/expand/BatchNor
                                                                  m[0][0]']

 expanded_conv_5/depthwise (Dep  (None, 28, 28, 120)  3000
['re_lu_12[0][0]']
 thwiseConv2D)

 expanded_conv_5/depthwise/Batc  (None, 28, 28, 120)  480
['expanded_conv_5/depthwise[0][0]
```

```
 hNorm (BatchNormalization)                                      ']

 re_lu_13 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_5/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 120)   0
['re_lu_13[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_5/squeeze_excite  (None, 1, 1, 32)    3872
['expanded_conv_5/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 32)    0
['expanded_conv_5/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 120)   3960
['expanded_conv_5/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']

 tf.__operators__.add_3 (TFOpLa  (None, 1, 1, 120)   0
['expanded_conv_5/squeeze_excite/
 mbda)                                                           Conv_1[0][0]']

 re_lu_14 (ReLU)                (None, 1, 1, 120)    0
['tf.__operators__.add_3[0][0]']

 tf.math.multiply_3 (TFOpLambda  (None, 1, 1, 120)   0
['re_lu_14[0][0]']
 )

 expanded_conv_5/squeeze_excite  (None, 28, 28, 120)  0
['re_lu_13[0][0]',
 /Mul (Multiply)
'tf.math.multiply_3[0][0]']

 expanded_conv_5/project (Conv2  (None, 28, 28, 40)  4800
['expanded_conv_5/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_5/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_5/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_5/Add (Add)      (None, 28, 28, 40)   0
```

```
['expanded_conv_4/Add[0][0]',
'expanded_conv_5/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_6/expand (Conv2D  (None, 28, 28, 240)  9600
['expanded_conv_5/Add[0][0]']
 )

 expanded_conv_6/expand/BatchNo  (None, 28, 28, 240)  960
['expanded_conv_6/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_4 (TFOpLa  (None, 28, 28, 240)  0
['expanded_conv_6/expand/BatchNor
 mbda)                                                           m[0][0]']

 re_lu_15 (ReLU)                (None, 28, 28, 240)  0
['tf.__operators__.add_4[0][0]']

 tf.math.multiply_4 (TFOpLambda  (None, 28, 28, 240)  0
['re_lu_15[0][0]']
 )

 multiply_1 (Multiply)          (None, 28, 28, 240)  0
['expanded_conv_6/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_4[0][0]']

 expanded_conv_6/depthwise/pad   (None, 29, 29, 240)  0
['multiply_1[0][0]']
 (ZeroPadding2D)

 expanded_conv_6/depthwise (Dep  (None, 14, 14, 240)  2160
['expanded_conv_6/depthwise/pad[0
 thwiseConv2D)                                                   ][0]']

 expanded_conv_6/depthwise/Batc  (None, 14, 14, 240)  960
['expanded_conv_6/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 tf.__operators__.add_5 (TFOpLa  (None, 14, 14, 240)  0
['expanded_conv_6/depthwise/Batch
 mbda)                                                           Norm[0][0]']

 re_lu_16 (ReLU)                (None, 14, 14, 240)  0
['tf.__operators__.add_5[0][0]']

 tf.math.multiply_5 (TFOpLambda  (None, 14, 14, 240)  0
['re_lu_16[0][0]']
 )

 multiply_2 (Multiply)          (None, 14, 14, 240)  0
['expanded_conv_6/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_5[0][0]']

 expanded_conv_6/project (Conv2  (None, 14, 14, 80)  19200
['multiply_2[0][0]']
 D)

 expanded_conv_6/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_6/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_7/expand (Conv2D  (None, 14, 14, 200)  16000
['expanded_conv_6/project/BatchNo
 )                                                               rm[0][0]']

 expanded_conv_7/expand/BatchNo  (None, 14, 14, 200)  800
['expanded_conv_7/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_6 (TFOpLa  (None, 14, 14, 200)  0
['expanded_conv_7/expand/BatchNor
 mbda)                                                           m[0][0]']

 re_lu_17 (ReLU)                (None, 14, 14, 200)  0
['tf.__operators__.add_6[0][0]']

 tf.math.multiply_6 (TFOpLambda  (None, 14, 14, 200)  0
['re_lu_17[0][0]']
 )

 multiply_3 (Multiply)          (None, 14, 14, 200)  0
['expanded_conv_7/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_6[0][0]']

 expanded_conv_7/depthwise (Dep  (None, 14, 14, 200)  1800
['multiply_3[0][0]']
 thwiseConv2D)

 expanded_conv_7/depthwise/Batc  (None, 14, 14, 200)  800
['expanded_conv_7/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']
 tf.__operators__.add_7 (TFOpLa  (None, 14, 14, 200)  0
['expanded_conv_7/depthwise/Batch
 mbda)                                                           Norm[0][0]']

 re_lu_18 (ReLU)                (None, 14, 14, 200)  0
['tf.__operators__.add_7[0][0]']

 tf.math.multiply_7 (TFOpLambda  (None, 14, 14, 200)  0
['re_lu_18[0][0]']
 )

 multiply_4 (Multiply)          (None, 14, 14, 200)  0
['expanded_conv_7/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_7[0][0]']

 expanded_conv_7/project (Conv2  (None, 14, 14, 80)  16000
['multiply_4[0][0]']
 D)

 expanded_conv_7/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_7/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_7/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_6/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_7/project/BatchNo

                                                                 rm[0][0]']

 expanded_conv_8/expand (Conv2D  (None, 14, 14, 184)  14720
['expanded_conv_7/Add[0][0]']
 )

 expanded_conv_8/expand/BatchNo  (None, 14, 14, 184)  736
['expanded_conv_8/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_8 (TFOpLa  (None, 14, 14, 184)  0
['expanded_conv_8/expand/BatchNor
 mbda)                                                           m[0][0]']

 re_lu_19 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_8[0][0]']

 tf.math.multiply_8 (TFOpLambda  (None, 14, 14, 184)  0
['re_lu_19[0][0]']
 )
 multiply_5 (Multiply)          (None, 14, 14, 184)  0
['expanded_conv_8/expand/BatchNor
                                                                m[0][0]',
'tf.math.multiply_8[0][0]']

 expanded_conv_8/depthwise (Dep  (None, 14, 14, 184)  1656
['multiply_5[0][0]']
 thwiseConv2D)

 expanded_conv_8/depthwise/Batc  (None, 14, 14, 184)  736
['expanded_conv_8/depthwise[0][0]
 hNorm (BatchNormalization)                                     ']

 tf.__operators__.add_9 (TFOpLa  (None, 14, 14, 184)  0
['expanded_conv_8/depthwise/Batch
 mbda)                                                          Norm[0][0]']

 re_lu_20 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_9[0][0]']

 tf.math.multiply_9 (TFOpLambda  (None, 14, 14, 184)  0
['re_lu_20[0][0]']
 )

 multiply_6 (Multiply)          (None, 14, 14, 184)  0
['expanded_conv_8/depthwise/Batch
                                                                Norm[0][0]',
'tf.math.multiply_9[0][0]']

 expanded_conv_8/project (Conv2  (None, 14, 14, 80)  14720
['multiply_6[0][0]']
 D)

 expanded_conv_8/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_8/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_8/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_7/Add[0][0]',
'expanded_conv_8/project/BatchNo
                                                                rm[0][0]']

 expanded_conv_9/expand (Conv2D  (None, 14, 14, 184)  14720
['expanded_conv_8/Add[0][0]']
 )

 expanded_conv_9/expand/BatchNo  (None, 14, 14, 184)  736
```

```
['expanded_conv_9/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_10 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_9/expand/BatchNor
 ambda)                                                          m[0][0]']

 re_lu_21 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_10[0][0]']

 tf.math.multiply_10 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_21[0][0]']
 a)

 multiply_7 (Multiply)          (None, 14, 14, 184)  0
['expanded_conv_9/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_10[0][0]']

 expanded_conv_9/depthwise (Dep  (None, 14, 14, 184)  1656
['multiply_7[0][0]']
 thwiseConv2D)

 expanded_conv_9/depthwise/Batc  (None, 14, 14, 184)  736
['expanded_conv_9/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 tf.__operators__.add_11 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_9/depthwise/Batch
 ambda)                                                          Norm[0][0]']

 re_lu_22 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_11[0][0]']

 tf.math.multiply_11 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_22[0][0]']
 a)

 multiply_8 (Multiply)          (None, 14, 14, 184)  0
['expanded_conv_9/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_11[0][0]']

 expanded_conv_9/project (Conv2  (None, 14, 14, 80)  14720
['multiply_8[0][0]']
 D)

 expanded_conv_9/project/BatchN  (None, 14, 14, 80)  320
```

```
['expanded_conv_9/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_9/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_8/Add[0][0]',
'expanded_conv_9/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_10/expand (Conv2  (None, 14, 14, 480)  38400
['expanded_conv_9/Add[0][0]']
 D)

 expanded_conv_10/expand/BatchN  (None, 14, 14, 480)  1920
['expanded_conv_10/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_12 (TFOpL  (None, 14, 14, 480)  0
['expanded_conv_10/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_23 (ReLU)                (None, 14, 14, 480)  0
['tf.__operators__.add_12[0][0]']

 tf.math.multiply_12 (TFOpLambd  (None, 14, 14, 480)  0
['re_lu_23[0][0]']
 a)

 multiply_9 (Multiply)          (None, 14, 14, 480)  0
['expanded_conv_10/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_12[0][0]']

 expanded_conv_10/depthwise (De  (None, 14, 14, 480)  4320
['multiply_9[0][0]']
 pthwiseConv2D)

 expanded_conv_10/depthwise/Bat  (None, 14, 14, 480)  1920
['expanded_conv_10/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_13 (TFOpL  (None, 14, 14, 480)  0
['expanded_conv_10/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_24 (ReLU)                (None, 14, 14, 480)  0
['tf.__operators__.add_13[0][0]']

 tf.math.multiply_13 (TFOpLambd  (None, 14, 14, 480)  0
['re_lu_24[0][0]']
 a)

 multiply_10 (Multiply)         (None, 14, 14, 480)  0
['expanded_conv_10/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_13[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 480)   0
['multiply_10[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_10/squeeze_excit  (None, 1, 1, 120)   57720
['expanded_conv_10/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 120)   0
['expanded_conv_10/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 480)   58080
['expanded_conv_10/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_14 (TFOpL  (None, 1, 1, 480)   0
['expanded_conv_10/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_25 (ReLU)                (None, 1, 1, 480)    0
['tf.__operators__.add_14[0][0]']

 tf.math.multiply_14 (TFOpLambd  (None, 1, 1, 480)   0
['re_lu_25[0][0]']
 a)

 expanded_conv_10/squeeze_excit  (None, 14, 14, 480)  0
['multiply_10[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_14[0][0]']

 expanded_conv_10/project (Conv  (None, 14, 14, 112)  53760
['expanded_conv_10/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_10/project/Batch  (None, 14, 14, 112)  448
['expanded_conv_10/project[0][0]'
```

```
 Norm (BatchNormalization)                                       ]

 expanded_conv_11/expand (Conv2  (None, 14, 14, 672)  75264
['expanded_conv_10/project/BatchN
 D)                                                              orm[0][0]']

 expanded_conv_11/expand/BatchN  (None, 14, 14, 672)  2688
['expanded_conv_11/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_15 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_11/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_26 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_15[0][0]']

 tf.math.multiply_15 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_26[0][0]']
 a)

 multiply_11 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_11/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_15[0][0]']

 expanded_conv_11/depthwise (De  (None, 14, 14, 672)  6048
['multiply_11[0][0]']
 pthwiseConv2D)

 expanded_conv_11/depthwise/Bat  (None, 14, 14, 672)  2688
['expanded_conv_11/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_16 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_11/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_27 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_16[0][0]']

 tf.math.multiply_16 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_27[0][0]']
 a)

 multiply_12 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_11/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_16[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 672)   0
['multiply_12[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_11/squeeze_excit  (None, 1, 1, 168)   113064
['expanded_conv_11/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 168)   0
['expanded_conv_11/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 672)   113568
['expanded_conv_11/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_17 (TFOpL  (None, 1, 1, 672)   0
['expanded_conv_11/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_28 (ReLU)                (None, 1, 1, 672)    0
['tf.__operators__.add_17[0][0]']

 tf.math.multiply_17 (TFOpLambd  (None, 1, 1, 672)   0
['re_lu_28[0][0]']
 a)

 expanded_conv_11/squeeze_excit  (None, 14, 14, 672)  0
['multiply_12[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_17[0][0]']

 expanded_conv_11/project (Conv  (None, 14, 14, 112)  75264
['expanded_conv_11/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_11/project/Batch  (None, 14, 14, 112)  448
['expanded_conv_11/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_11/Add (Add)     (None, 14, 14, 112)  0
['expanded_conv_10/project/BatchN
                                                                 orm[0][0]',
'expanded_conv_11/project/BatchN
```

```
                                                                 orm[0][0]']

 expanded_conv_12/expand (Conv2  (None, 14, 14, 672)  75264
['expanded_conv_11/Add[0][0]']
 D)

 expanded_conv_12/expand/BatchN  (None, 14, 14, 672)  2688
['expanded_conv_12/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_18 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_12/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_29 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_18[0][0]']

 tf.math.multiply_18 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_29[0][0]']
 a)

 multiply_13 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_12/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_18[0][0]']

 expanded_conv_12/depthwise/pad  (None, 17, 17, 672)  0
['multiply_13[0][0]']
  (ZeroPadding2D)

 expanded_conv_12/depthwise (De  (None, 7, 7, 672)   16800
['expanded_conv_12/depthwise/pad[
 pthwiseConv2D)                                                  0][0]']

 expanded_conv_12/depthwise/Bat  (None, 7, 7, 672)   2688
['expanded_conv_12/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_19 (TFOpL  (None, 7, 7, 672)   0
['expanded_conv_12/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_30 (ReLU)                (None, 7, 7, 672)    0
['tf.__operators__.add_19[0][0]']

 tf.math.multiply_19 (TFOpLambd  (None, 7, 7, 672)   0
['re_lu_30[0][0]']
 a)
 multiply_14 (Multiply)         (None, 7, 7, 672)    0
['expanded_conv_12/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_19[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 672)   0
['multiply_14[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_12/squeeze_excit  (None, 1, 1, 168)   113064
['expanded_conv_12/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 168)   0
['expanded_conv_12/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 672)   113568
['expanded_conv_12/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_20 (TFOpL  (None, 1, 1, 672)   0
['expanded_conv_12/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_31 (ReLU)                (None, 1, 1, 672)    0
['tf.__operators__.add_20[0][0]']

 tf.math.multiply_20 (TFOpLambd  (None, 1, 1, 672)   0
['re_lu_31[0][0]']
 a)

 expanded_conv_12/squeeze_excit  (None, 7, 7, 672)   0
['multiply_14[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_20[0][0]']

 expanded_conv_12/project (Conv  (None, 7, 7, 160)   107520
['expanded_conv_12/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_12/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_12/project[0][0]'
 Norm (BatchNormalization)                                       ]
 expanded_conv_13/expand (Conv2  (None, 7, 7, 960)   153600
['expanded_conv_12/project/BatchN
 D)                                                              orm[0][0]']

 expanded_conv_13/expand/BatchN  (None, 7, 7, 960)   3840
['expanded_conv_13/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_21 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_13/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_32 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_21[0][0]']

 tf.math.multiply_21 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_32[0][0]']
 a)

 multiply_15 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_13/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_21[0][0]']

 expanded_conv_13/depthwise (De  (None, 7, 7, 960)   24000
['multiply_15[0][0]']
 pthwiseConv2D)

 expanded_conv_13/depthwise/Bat  (None, 7, 7, 960)   3840
['expanded_conv_13/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_22 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_13/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_33 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_22[0][0]']

 tf.math.multiply_22 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_33[0][0]']
 a)

 multiply_16 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_13/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_22[0][0]']
```

```
 expanded_conv_13/squeeze_excit  (None, 1, 1, 960)   0
['multiply_16[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_13/squeeze_excit  (None, 1, 1, 240)   230640
['expanded_conv_13/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_13/squeeze_excit  (None, 1, 1, 240)   0
['expanded_conv_13/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_13/squeeze_excit  (None, 1, 1, 960)   231360
['expanded_conv_13/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_23 (TFOpL  (None, 1, 1, 960)   0
['expanded_conv_13/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_34 (ReLU)                (None, 1, 1, 960)    0
['tf.__operators__.add_23[0][0]']

 tf.math.multiply_23 (TFOpLambd  (None, 1, 1, 960)   0
['re_lu_34[0][0]']
 a)

 expanded_conv_13/squeeze_excit  (None, 7, 7, 960)   0
['multiply_16[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_23[0][0]']

 expanded_conv_13/project (Conv  (None, 7, 7, 160)   153600
['expanded_conv_13/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_13/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_13/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_13/Add (Add)     (None, 7, 7, 160)    0
['expanded_conv_12/project/BatchN
                                                                 orm[0][0]',
'expanded_conv_13/project/BatchN
                                                                 orm[0][0]']
 expanded_conv_14/expand (Conv2  (None, 7, 7, 960)   153600
['expanded_conv_13/Add[0][0]']
 D)

 expanded_conv_14/expand/BatchN  (None, 7, 7, 960)   3840
['expanded_conv_14/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_24 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_14/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_35 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_24[0][0]']

 tf.math.multiply_24 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_35[0][0]']
 a)

 multiply_17 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_14/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_24[0][0]']

 expanded_conv_14/depthwise (De  (None, 7, 7, 960)   24000
['multiply_17[0][0]']
 pthwiseConv2D)

 expanded_conv_14/depthwise/Bat  (None, 7, 7, 960)   3840
['expanded_conv_14/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_25 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_14/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_36 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_25[0][0]']

 tf.math.multiply_25 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_36[0][0]']
 a)

 multiply_18 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_14/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_25[0][0]']
```

```
 expanded_conv_14/squeeze_excit  (None, 1, 1, 960)   0
['multiply_18[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_14/squeeze_excit  (None, 1, 1, 240)   230640
['expanded_conv_14/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_14/squeeze_excit  (None, 1, 1, 240)   0
['expanded_conv_14/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_14/squeeze_excit  (None, 1, 1, 960)   231360
['expanded_conv_14/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_26 (TFOpL  (None, 1, 1, 960)   0
['expanded_conv_14/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_37 (ReLU)                (None, 1, 1, 960)    0
['tf.__operators__.add_26[0][0]']

 tf.math.multiply_26 (TFOpLambd  (None, 1, 1, 960)   0
['re_lu_37[0][0]']
 a)

 expanded_conv_14/squeeze_excit  (None, 7, 7, 960)   0
['multiply_18[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_26[0][0]']

 expanded_conv_14/project (Conv  (None, 7, 7, 160)   153600
['expanded_conv_14/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_14/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_14/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_14/Add (Add)     (None, 7, 7, 160)    0
['expanded_conv_13/Add[0][0]',
'expanded_conv_14/project/BatchN
                                                                 orm[0][0]']

 Conv_1 (Conv2D)                (None, 7, 7, 960)    153600
```

```
['expanded_conv_14/Add[0][0]']

 Conv_1/BatchNorm (BatchNormali  (None, 7, 7, 960)   3840
['Conv_1[0][0]']
 zation)

 tf.__operators__.add_27 (TFOpL  (None, 7, 7, 960)   0
['Conv_1/BatchNorm[0][0]']
 ambda)

 re_lu_38 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_27[0][0]']

 tf.math.multiply_27 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_38[0][0]']
 a)

 multiply_19 (Multiply)         (None, 7, 7, 960)    0
['Conv_1/BatchNorm[0][0]',
'tf.math.multiply_27[0][0]']

 global_average_pooling2d_1 (Gl  (None, 1, 1, 960)   0
['multiply_19[0][0]']
 obalAveragePooling2D)

 Conv_2 (Conv2D)                (None, 1, 1, 1280)   1230080
['global_average_pooling2d_1[0][0
                                                                 ]']

 tf.__operators__.add_28 (TFOpL  (None, 1, 1, 1280)  0
['Conv_2[0][0]']
 ambda)

 re_lu_39 (ReLU)                (None, 1, 1, 1280)   0
['tf.__operators__.add_28[0][0]']

 tf.math.multiply_28 (TFOpLambd  (None, 1, 1, 1280)  0
['re_lu_39[0][0]']
 a)

 multiply_20 (Multiply)         (None, 1, 1, 1280)   0
['Conv_2[0][0]',
'tf.math.multiply_28[0][0]']

 dropout (Dropout)              (None, 1, 1, 1280)   0
['multiply_20[0][0]']

 Logits (Conv2D)                (None, 1, 1, 1000)   1281000
```

```
['dropout[0][0]']

 flatten (Flatten)              (None, 1000)         0
['Logits[0][0]']

 Predictions (Activation)       (None, 1000)         0
['flatten[0][0]']

================================================================================
==================
Total params: 5,507,432
Trainable params: 5,483,032
Non-trainable params: 24,400
--------------------------------------------------------------------------------
------------------
```

[ ]:

```
# Call back 1:
base_learning_rate = 1e-4
opt1 = tf.keras.optimizers.Adam(learning_rate=base_learning_rate)
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='accuracy', min_delta=0, patience=4, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

# Call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1, callback_2]

#compiling our Model for dataset
mobilev3model.compile(optimizer=opt1,
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = mobilev3model.fit(
    train_mobilenet,
    epochs=60,
    validation_data=test_mobilenet,
    class_weight=class_weights,
    callbacks=callback_list)
```

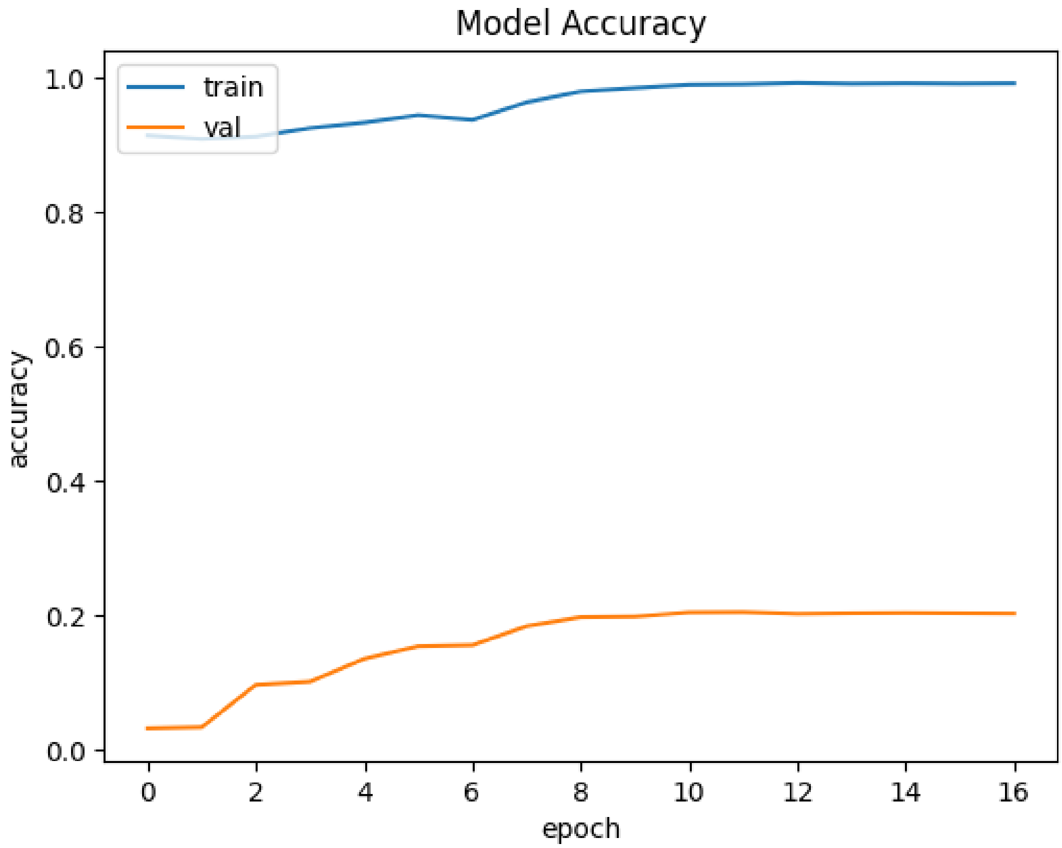
```
Epoch 1/60
```

```
299/299 [==============================] - 48s 60ms/step - loss: 0.4436 -
accuracy: 0.9143 - val_loss: 6.7276 - val_accuracy: 0.0331 - lr: 1.0000e-04
Epoch 2/60
299/299 [==============================] - 17s 57ms/step - loss: 0.4489 -
accuracy: 0.9089 - val_loss: 5.5325 - val_accuracy: 0.0347 - lr: 1.0000e-04
Epoch 3/60
299/299 [==============================] - 17s 57ms/step - loss: 0.4534 -
accuracy: 0.9127 - val_loss: 4.7189 - val_accuracy: 0.0979 - lr: 1.0000e-04
Epoch 4/60
299/299 [==============================] - 17s 57ms/step - loss: 0.3895 -
accuracy: 0.9252 - val_loss: 5.1645 - val_accuracy: 0.1025 - lr: 1.0000e-04
Epoch 5/60
299/299 [==============================] - 17s 57ms/step - loss: 0.3549 -
accuracy: 0.9335 - val_loss: 4.8629 - val_accuracy: 0.1364 - lr: 1.0000e-04
Epoch 6/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2848 -
accuracy: 0.9445 - val_loss: 5.5299 - val_accuracy: 0.1552 - lr: 1.0000e-04
Epoch 7/60
299/299 [==============================] - 17s 57ms/step - loss: 0.3093 -
accuracy: 0.9375 - val_loss: 6.6987 - val_accuracy: 0.1569 - lr: 1.0000e-04
Epoch 8/60
299/299 [==============================] - 17s 57ms/step - loss: 0.1998 -
accuracy: 0.9637 - val_loss: 5.6490 - val_accuracy: 0.1849 - lr: 1.0000e-05
Epoch 9/60
299/299 [==============================] - 17s 57ms/step - loss: 0.1178 -
accuracy: 0.9797 - val_loss: 5.3925 - val_accuracy: 0.1983 - lr: 1.0000e-05
Epoch 10/60
299/299 [==============================] - 17s 57ms/step - loss: 0.0946 -
accuracy: 0.9849 - val_loss: 5.3837 - val_accuracy: 0.1992 - lr: 1.0000e-05
Epoch 11/60
299/299 [==============================] - 17s 57ms/step - loss: 0.0709 -
accuracy: 0.9896 - val_loss: 5.4038 - val_accuracy: 0.2054 - lr: 1.0000e-05
Epoch 12/60
299/299 [==============================] - 17s 57ms/step - loss: 0.0652 -
accuracy: 0.9904 - val_loss: 5.4103 - val_accuracy: 0.2059 - lr: 1.0000e-06
Epoch 13/60
299/299 [==============================] - 17s 57ms/step - loss: 0.0567 -
accuracy: 0.9924 - val_loss: 5.4310 - val_accuracy: 0.2033 - lr: 1.0000e-06
Epoch 14/60
299/299 [==============================] - 17s 56ms/step - loss: 0.0601 -
accuracy: 0.9912 - val_loss: 5.4518 - val_accuracy: 0.2042 - lr: 1.0000e-06
Epoch 15/60
299/299 [==============================] - 17s 57ms/step - loss: 0.0573 -
accuracy: 0.9917 - val_loss: 5.4706 - val_accuracy: 0.2046 - lr: 1.0000e-06
Epoch 16/60
299/299 [==============================] - 17s 57ms/step - loss: 0.0601 -
accuracy: 0.9913 - val_loss: 5.4850 - val_accuracy: 0.2042 - lr: 1.0000e-07
Epoch 17/60
```

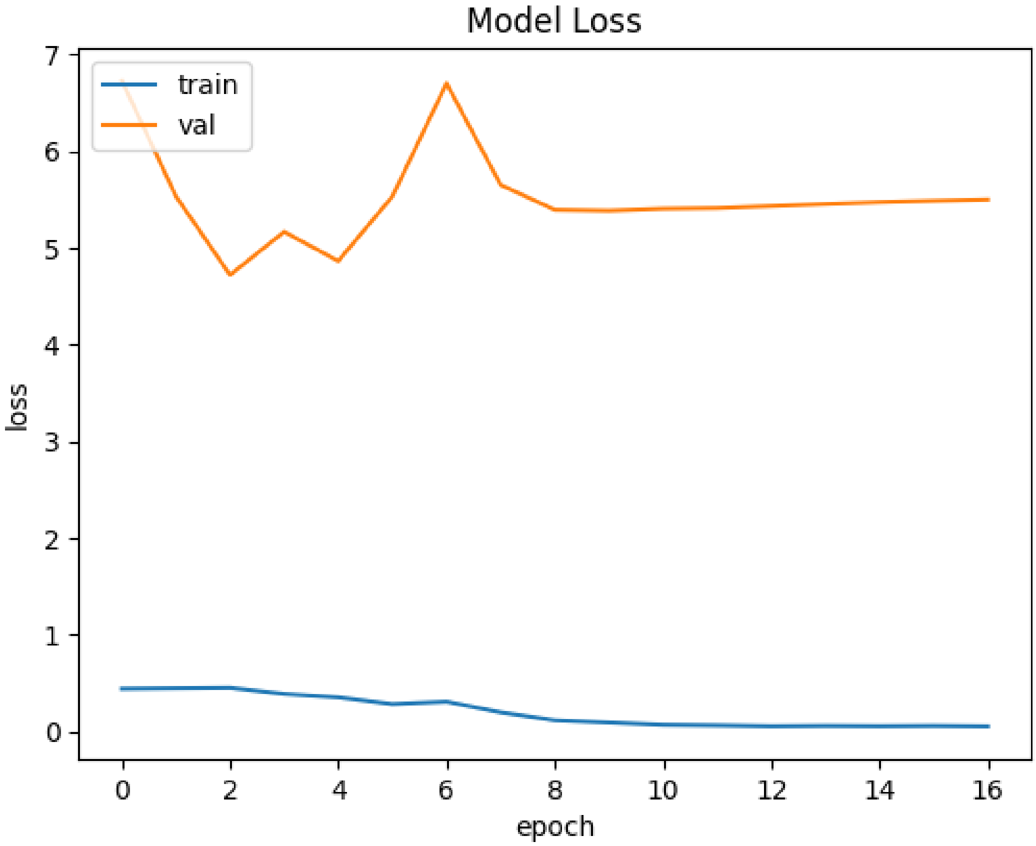
```
299/299 [==============================] - 17s 57ms/step - loss: 0.0555 -
accuracy: 0.9917 - val_loss: 5.4980 - val_accuracy: 0.2038 - lr: 1.0000e-07
```

Training was ran three consecutive times, and stopped early on all three runs; approximately 70 epochs in total. This is the highest validation accuracy that could be achieved.

```
[ ]: plt.plot(history.history['accuracy'])
     plt.plot(history.history['val_accuracy'])
     plt.title('Model Accuracy')
     plt.ylabel('accuracy')
     plt.xlabel('epoch')
     plt.legend(['train', 'val'], loc='upper left')
     plt.show()
```



```
[ ]: plt.plot(history.history['loss'])
     plt.plot(history.history['val_loss'])
     plt.title('Model Loss')
     plt.ylabel('loss')
     plt.xlabel('epoch')
     plt.legend(['train', 'val'], loc='upper left')
     plt.show()
```



```
[ ]: mobilev3model.save(os.path.join(SAVE_DIR,'mobileNetV3_noPretrain_noTuning.h5'))
```

1.11 Train MobileNetV2 with Filtered Dataset and Data Augmentation, 128 Batch Size, Pretrained on ImageNet

```
[ ]: import os
     import shutil
     from google.colab import drive
     import tensorflow as tf
     from matplotlib import pyplot as plt

     DIRPATH = '/content/VMMRdb_LrgImgCount/'

     drive.mount('/content/drive/')
     shutil.unpack_archive("/content/drive/MyDrive/endg511project/VMMRdb_LrgImgCount.
      ↪zip", DIRPATH)

     num_classes = len(os.listdir(DIRPATH))
     print(num_classes)
```

```
Drive already mounted at /content/drive/; to attempt to forcibly remount, call
drive.mount("/content/drive/", force_remount=True).
25
```

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

# Data augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    horizontal_flip=True,
    validation_split=0.2)

# Training set
train_generator = train_datagen.flow_from_directory(
    DIRPATH,
    target_size=(224, 224),
    batch_size=128,
    class_mode='sparse',
    subset='training')

# Validation set
validation_generator = train_datagen.flow_from_directory(
    DIRPATH,
    target_size=(224, 224),
    batch_size=128,
    class_mode='sparse',
    subset='validation')
```

```
Found 9572 images belonging to 25 classes.
Found 2379 images belonging to 25 classes.
```

[ ]:
```
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Model

# Define MobileNetV2 model with pre-trained weights and excluding the final␣
 ↪classification layer
mobilev2model = tf.keras.applications.MobileNetV2(
    input_shape=(224,224,3),
    alpha=1.0,
    include_top=False,
    weights='imagenet',
    input_tensor=None,
    pooling='max'
)

# Define new output layer with softmax activation for `num_classes` classes
output_layer = Dense(num_classes, activation='softmax')(mobilev2model.output)

# Create a new model with the same input as the MobileNetV2 model and the new␣
  ↪output layer
mobilev2model = Model(inputs=mobilev2model.input, outputs=output_layer)

# Print model summary
mobilev2model.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applicatio
ns/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h
5
9406464/9406464 [==============================] - 1s 0us/step
Model: "model"
_________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
=================================================================================
==================
 input_3 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864
['input_3[0][0]']

                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0][0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0
['bn_Conv1[0][0]']

                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288
['Conv1_relu[0][0]']
 wiseConv2D)                    )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128
['expanded_conv_depthwise[0][0]']
 tchNormalization)              )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0
['expanded_conv_depthwise_BN[0][0
```

```
 ReLU)                          )                                ]']

 expanded_conv_project (Conv2D)  (None, 112, 112, 16  512
['expanded_conv_depthwise_relu[0]
                                )                                [0]']

 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64
['expanded_conv_project[0][0]']
 hNormalization)                )

 block_1_expand (Conv2D)        (None, 112, 112, 96  1536
['expanded_conv_project_BN[0][0]'
                                )                                ]

 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384
['block_1_expand[0][0]']
 ization)                       )

 block_1_expand_relu (ReLU)     (None, 112, 112, 96  0
['block_1_expand_BN[0][0]']
                                )

 block_1_pad (ZeroPadding2D)    (None, 113, 113, 96  0
['block_1_expand_relu[0][0]']
                                )

 block_1_depthwise (DepthwiseCo  (None, 56, 56, 96)  864
['block_1_pad[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 56, 56, 96)  384
['block_1_depthwise[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 56, 56, 96)   0
['block_1_depthwise_BN[0][0]']

 block_1_project (Conv2D)       (None, 56, 56, 24)   2304
['block_1_depthwise_relu[0][0]']

 block_1_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_1_project[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_1_project_BN[0][0]']

 block_2_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_2_expand[0][0]']
 ization)

 block_2_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_2_expand_BN[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 56, 56, 144)  1296
['block_2_expand_relu[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 56, 56, 144)  576
['block_2_depthwise[0][0]']
 malization)

 block_2_depthwise_relu (ReLU)  (None, 56, 56, 144)  0
['block_2_depthwise_BN[0][0]']

 block_2_project (Conv2D)       (None, 56, 56, 24)   3456
['block_2_depthwise_relu[0][0]']

 block_2_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_2_project[0][0]']
 lization)

 block_2_add (Add)              (None, 56, 56, 24)   0
['block_1_project_BN[0][0]',
'block_2_project_BN[0][0]']

 block_3_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_2_add[0][0]']

 block_3_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_3_expand[0][0]']
 ization)

 block_3_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_3_expand_BN[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 57, 57, 144)  0
['block_3_expand_relu[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 28, 28, 144)  1296
['block_3_pad[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 28, 28, 144)  576
['block_3_depthwise[0][0]']
 malization)
 block_3_depthwise_relu (ReLU)  (None, 28, 28, 144)  0
['block_3_depthwise_BN[0][0]']

 block_3_project (Conv2D)       (None, 28, 28, 32)   4608
['block_3_depthwise_relu[0][0]']

 block_3_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_3_project[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_3_project_BN[0][0]']

 block_4_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_4_expand[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_4_expand_BN[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_4_expand_relu[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_4_depthwise[0][0]']
 malization)

 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_4_depthwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 28, 28, 32)   6144
['block_4_depthwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_4_project[0][0]']
 lization)

 block_4_add (Add)              (None, 28, 28, 32)   0
['block_3_project_BN[0][0]',
'block_4_project_BN[0][0]']

 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_4_add[0][0]']

 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_5_expand[0][0]']
```

```
 ization)

 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_5_expand_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_5_expand_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_5_depthwise[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_5_depthwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 28, 28, 32)   6144
['block_5_depthwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_5_project[0][0]']
 lization)

 block_5_add (Add)              (None, 28, 28, 32)   0
['block_4_add[0][0]',
'block_5_project_BN[0][0]']

 block_6_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_5_add[0][0]']

 block_6_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_6_expand[0][0]']
 ization)

 block_6_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_6_expand_BN[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 29, 29, 192)  0
['block_6_expand_relu[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 14, 14, 192)  1728
['block_6_pad[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 14, 14, 192)  768
['block_6_depthwise[0][0]']
 malization)
 block_6_depthwise_relu (ReLU)  (None, 14, 14, 192)  0
['block_6_depthwise_BN[0][0]']

 block_6_project (Conv2D)       (None, 14, 14, 64)   12288
['block_6_depthwise_relu[0][0]']

 block_6_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_6_project[0][0]']
 lization)

 block_7_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_6_project_BN[0][0]']

 block_7_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_7_expand[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_7_expand_BN[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_7_expand_relu[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_7_depthwise[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_7_depthwise_BN[0][0]']

 block_7_project (Conv2D)       (None, 14, 14, 64)   24576
['block_7_depthwise_relu[0][0]']

 block_7_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_7_project[0][0]']
 lization)

 block_7_add (Add)              (None, 14, 14, 64)   0
['block_6_project_BN[0][0]',
'block_7_project_BN[0][0]']

 block_8_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_7_add[0][0]']

 block_8_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_8_expand[0][0]']
 ization)
 block_8_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_8_expand_BN[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_8_expand_relu[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_8_depthwise[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_8_depthwise_BN[0][0]']

 block_8_project (Conv2D)       (None, 14, 14, 64)   24576
['block_8_depthwise_relu[0][0]']

 block_8_project_BN (BatchNorma  (None, 14, 14, 64)   256
['block_8_project[0][0]']
 lization)

 block_8_add (Add)              (None, 14, 14, 64)   0
['block_7_add[0][0]',
'block_8_project_BN[0][0]']

 block_9_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_8_add[0][0]']

 block_9_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_9_expand[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_9_expand_BN[0][0]']

 block_9_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_9_expand_relu[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_9_depthwise[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_9_depthwise_BN[0][0]']

 block_9_project (Conv2D)       (None, 14, 14, 64)   24576
```

```
['block_9_depthwise_relu[0][0]']

 block_9_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_9_project[0][0]']
 lization)

 block_9_add (Add)              (None, 14, 14, 64)   0
['block_8_add[0][0]',
'block_9_project_BN[0][0]']

 block_10_expand (Conv2D)       (None, 14, 14, 384)  24576
['block_9_add[0][0]']

 block_10_expand_BN (BatchNorma  (None, 14, 14, 384)  1536
['block_10_expand[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 14, 14, 384)  0
['block_10_expand_BN[0][0]']

 block_10_depthwise (DepthwiseC  (None, 14, 14, 384)  3456
['block_10_expand_relu[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 14, 14, 384)  1536
['block_10_depthwise[0][0]']
 rmalization)

 block_10_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_10_depthwise_BN[0][0]']

 block_10_project (Conv2D)      (None, 14, 14, 96)   36864
['block_10_depthwise_relu[0][0]']

 block_10_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_10_project[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_10_project_BN[0][0]']

 block_11_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_11_expand[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_11_expand_BN[0][0]']
```

```
 block_11_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_11_expand_relu[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_11_depthwise[0][0]']
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
['block_11_depthwise_BN[0][0]']

 block_11_project (Conv2D)      (None, 14, 14, 96)   55296
['block_11_depthwise_relu[0][0]']

 block_11_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_11_project[0][0]']
 alization)

 block_11_add (Add)             (None, 14, 14, 96)   0
['block_10_project_BN[0][0]',
'block_11_project_BN[0][0]']

 block_12_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_11_add[0][0]']

 block_12_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_12_expand[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_12_expand_BN[0][0]']

 block_12_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_12_expand_relu[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_12_depthwise[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
['block_12_depthwise_BN[0][0]']

 block_12_project (Conv2D)      (None, 14, 14, 96)   55296
['block_12_depthwise_relu[0][0]']

 block_12_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_12_project[0][0]']
```

```
 alization)

 block_12_add (Add)            (None, 14, 14, 96)   0
['block_11_add[0][0]',
'block_12_project_BN[0][0]']

 block_13_expand (Conv2D)      (None, 14, 14, 576)  55296
['block_12_add[0][0]']

 block_13_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_13_expand[0][0]']
 lization)

 block_13_expand_relu (ReLU)   (None, 14, 14, 576)  0
['block_13_expand_BN[0][0]']

 block_13_pad (ZeroPadding2D)  (None, 15, 15, 576)  0
['block_13_expand_relu[0][0]']

 block_13_depthwise (DepthwiseC  (None, 7, 7, 576)   5184
['block_13_pad[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304
['block_13_depthwise[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 7, 7, 576)   0
['block_13_depthwise_BN[0][0]']

 block_13_project (Conv2D)     (None, 7, 7, 160)    92160
['block_13_depthwise_relu[0][0]']

 block_13_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_13_project[0][0]']
 alization)

 block_14_expand (Conv2D)      (None, 7, 7, 960)    153600
['block_13_project_BN[0][0]']

 block_14_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_14_expand[0][0]']
 lization)

 block_14_expand_relu (ReLU)   (None, 7, 7, 960)    0
['block_14_expand_BN[0][0]']

 block_14_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_14_expand_relu[0][0]']
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_14_depthwise[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_14_depthwise_BN[0][0]']

 block_14_project (Conv2D)      (None, 7, 7, 160)    153600
['block_14_depthwise_relu[0][0]']

 block_14_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_14_project[0][0]']
 alization)

 block_14_add (Add)             (None, 7, 7, 160)    0
['block_13_project_BN[0][0]',
'block_14_project_BN[0][0]']

 block_15_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_14_add[0][0]']

 block_15_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_15_expand[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_15_expand_BN[0][0]']

 block_15_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_15_expand_relu[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_15_depthwise[0][0]']
 rmalization)

 block_15_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_15_depthwise_BN[0][0]']

 block_15_project (Conv2D)      (None, 7, 7, 160)    153600
['block_15_depthwise_relu[0][0]']

 block_15_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_15_project[0][0]']
 alization)
 block_15_add (Add)             (None, 7, 7, 160)    0
['block_14_add[0][0]',
'block_15_project_BN[0][0]']

 block_16_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_15_add[0][0]']

 block_16_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_16_expand[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_16_expand_BN[0][0]']

 block_16_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_16_expand_relu[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_16_depthwise[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_16_depthwise_BN[0][0]']

 block_16_project (Conv2D)      (None, 7, 7, 320)    307200
['block_16_depthwise_relu[0][0]']

 block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280
['block_16_project[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600
['block_16_project_BN[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120
['Conv_1[0][0]']

 out_relu (ReLU)                (None, 7, 7, 1280)   0
['Conv_1_bn[0][0]']

 global_max_pooling2d (GlobalMa  (None, 1280)        0
['out_relu[0][0]']
 xPooling2D)

 dense (Dense)                  (None, 25)           32025
['global_max_pooling2d[0][0]']
```

```
================================================================================
==================
Total params: 2,290,009
Trainable params: 2,255,897
Non-trainable params: 34,112
--------------------------------------------------------------------------------
------------------
```

[ ]:

```
# Call back 1:
base_learning_rate = 1e-5
opt1 = tf.keras.optimizers.Adam(learning_rate=base_learning_rate)
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='val_accuracy', min_delta=0, patience=4, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

# Call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1, callback_2]

#compiling our Model for dataset
mobilev2model.compile(optimizer=opt1,
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = mobilev2model.fit(
    train_generator,
    epochs=60,
    validation_data=validation_generator,
    callbacks=callback_list)
```

```
Epoch 1/60
75/75 [==============================] - 185s 2s/step - loss: 8.2733 - accuracy:
0.0543 - val_loss: 8.0098 - val_accuracy: 0.0597 - lr: 1.0000e-05
Epoch 2/60
75/75 [==============================] - 157s 2s/step - loss: 5.7164 - accuracy:
0.0755 - val_loss: 7.5849 - val_accuracy: 0.0647 - lr: 1.0000e-05
Epoch 3/60
75/75 [==============================] - 156s 2s/step - loss: 5.0202 - accuracy:
0.1076 - val_loss: 7.0076 - val_accuracy: 0.0744 - lr: 1.0000e-05
```

```
Epoch 4/60
75/75 [==============================] - 156s 2s/step - loss: 4.5388 - accuracy:
0.1347 - val_loss: 6.6553 - val_accuracy: 0.0769 - lr: 1.0000e-05
Epoch 5/60
75/75 [==============================] - 157s 2s/step - loss: 4.1696 - accuracy:
0.1688 - val_loss: 6.1572 - val_accuracy: 0.0862 - lr: 1.0000e-05
Epoch 6/60
75/75 [==============================] - 157s 2s/step - loss: 3.9135 - accuracy:
0.1890 - val_loss: 5.7270 - val_accuracy: 0.0971 - lr: 1.0000e-05
Epoch 7/60
75/75 [==============================] - 156s 2s/step - loss: 3.6355 - accuracy:
0.2142 - val_loss: 5.5760 - val_accuracy: 0.1021 - lr: 1.0000e-05
Epoch 8/60
75/75 [==============================] - 157s 2s/step - loss: 3.4510 - accuracy:
0.2385 - val_loss: 5.3284 - val_accuracy: 0.1114 - lr: 1.0000e-05
Epoch 9/60
75/75 [==============================] - 156s 2s/step - loss: 3.2489 - accuracy:
0.2556 - val_loss: 5.0230 - val_accuracy: 0.1324 - lr: 1.0000e-05
Epoch 10/60
75/75 [==============================] - 156s 2s/step - loss: 3.0594 - accuracy:
0.2809 - val_loss: 4.7956 - val_accuracy: 0.1383 - lr: 1.0000e-05
Epoch 11/60
75/75 [==============================] - 157s 2s/step - loss: 2.9531 - accuracy:
0.3013 - val_loss: 4.5109 - val_accuracy: 0.1602 - lr: 1.0000e-05
Epoch 12/60
75/75 [==============================] - 157s 2s/step - loss: 2.8320 - accuracy:
0.3111 - val_loss: 4.3522 - val_accuracy: 0.1686 - lr: 1.0000e-05
Epoch 13/60
75/75 [==============================] - 156s 2s/step - loss: 2.6915 - accuracy:
0.3293 - val_loss: 4.1161 - val_accuracy: 0.1799 - lr: 1.0000e-05
Epoch 14/60
75/75 [==============================] - 156s 2s/step - loss: 2.5742 - accuracy:
0.3444 - val_loss: 4.0112 - val_accuracy: 0.2051 - lr: 1.0000e-05
Epoch 15/60
75/75 [==============================] - 156s 2s/step - loss: 2.4881 - accuracy:
0.3645 - val_loss: 3.8933 - val_accuracy: 0.2064 - lr: 1.0000e-05
Epoch 16/60
75/75 [==============================] - 156s 2s/step - loss: 2.3975 - accuracy:
0.3726 - val_loss: 3.7758 - val_accuracy: 0.2110 - lr: 1.0000e-05
Epoch 17/60
75/75 [==============================] - 157s 2s/step - loss: 2.3330 - accuracy:
0.3844 - val_loss: 3.7160 - val_accuracy: 0.2186 - lr: 1.0000e-05
Epoch 18/60
75/75 [==============================] - 156s 2s/step - loss: 2.2259 - accuracy:
0.4010 - val_loss: 3.5450 - val_accuracy: 0.2228 - lr: 1.0000e-05
Epoch 19/60
75/75 [==============================] - 156s 2s/step - loss: 2.1656 - accuracy:
0.4164 - val_loss: 3.5301 - val_accuracy: 0.2358 - lr: 1.0000e-05
```

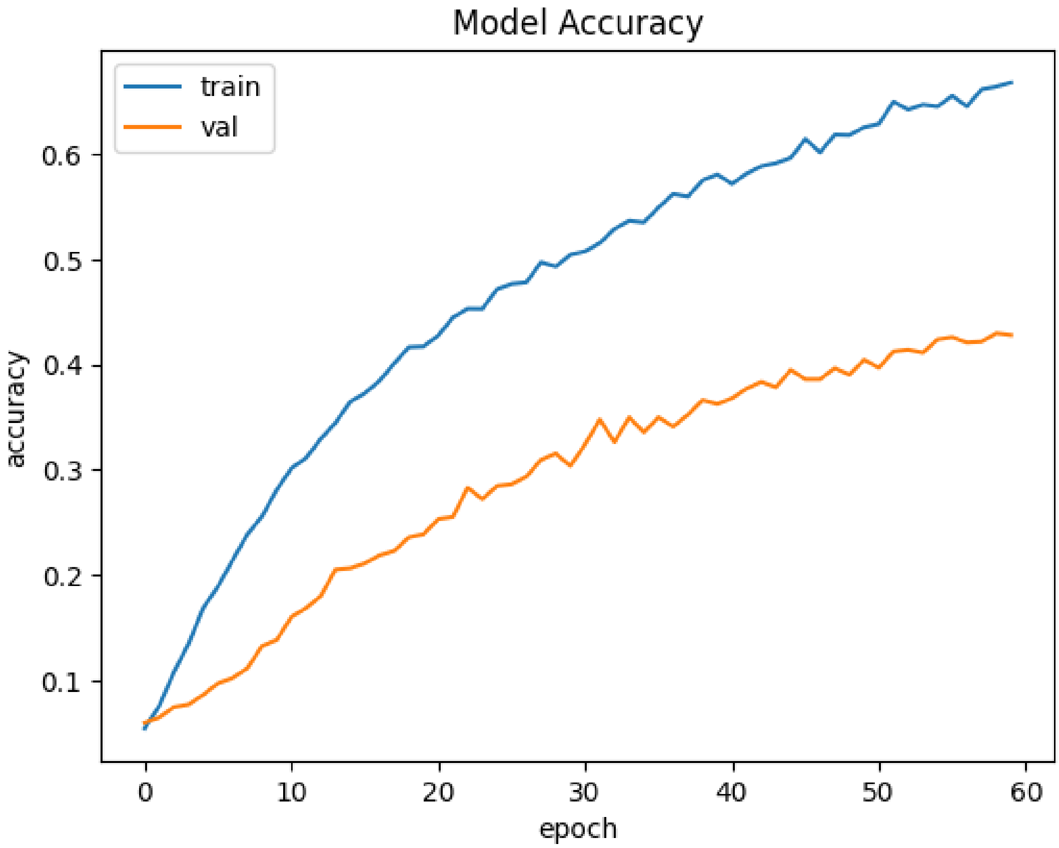
```
Epoch 20/60
75/75 [==============================] - 157s 2s/step - loss: 2.1249 - accuracy:
0.4170 - val_loss: 3.4421 - val_accuracy: 0.2388 - lr: 1.0000e-05
Epoch 21/60
75/75 [==============================] - 157s 2s/step - loss: 2.0598 - accuracy:
0.4275 - val_loss: 3.3540 - val_accuracy: 0.2530 - lr: 1.0000e-05
Epoch 22/60
75/75 [==============================] - 156s 2s/step - loss: 1.9829 - accuracy:
0.4445 - val_loss: 3.2011 - val_accuracy: 0.2551 - lr: 1.0000e-05
Epoch 23/60
75/75 [==============================] - 156s 2s/step - loss: 1.9141 - accuracy:
0.4529 - val_loss: 3.1859 - val_accuracy: 0.2829 - lr: 1.0000e-05
Epoch 24/60
75/75 [==============================] - 156s 2s/step - loss: 1.8959 - accuracy:
0.4528 - val_loss: 3.0738 - val_accuracy: 0.2720 - lr: 1.0000e-05
Epoch 25/60
75/75 [==============================] - 157s 2s/step - loss: 1.8181 - accuracy:
0.4715 - val_loss: 3.0252 - val_accuracy: 0.2846 - lr: 1.0000e-05
Epoch 26/60
75/75 [==============================] - 157s 2s/step - loss: 1.7746 - accuracy:
0.4765 - val_loss: 2.9680 - val_accuracy: 0.2863 - lr: 1.0000e-05
Epoch 27/60
75/75 [==============================] - 157s 2s/step - loss: 1.7439 - accuracy:
0.4780 - val_loss: 2.8898 - val_accuracy: 0.2934 - lr: 1.0000e-05
Epoch 28/60
75/75 [==============================] - 156s 2s/step - loss: 1.6851 - accuracy:
0.4971 - val_loss: 2.8494 - val_accuracy: 0.3094 - lr: 1.0000e-05
Epoch 29/60
75/75 [==============================] - 156s 2s/step - loss: 1.6560 - accuracy:
0.4930 - val_loss: 2.7636 - val_accuracy: 0.3153 - lr: 1.0000e-05
Epoch 30/60
75/75 [==============================] - 156s 2s/step - loss: 1.6242 - accuracy:
0.5042 - val_loss: 2.7732 - val_accuracy: 0.3035 - lr: 1.0000e-05
Epoch 31/60
75/75 [==============================] - 156s 2s/step - loss: 1.5861 - accuracy:
0.5073 - val_loss: 2.7376 - val_accuracy: 0.3241 - lr: 1.0000e-05
Epoch 32/60
75/75 [==============================] - 156s 2s/step - loss: 1.5413 - accuracy:
0.5154 - val_loss: 2.6136 - val_accuracy: 0.3476 - lr: 1.0000e-05
Epoch 33/60
75/75 [==============================] - 157s 2s/step - loss: 1.5075 - accuracy:
0.5286 - val_loss: 2.6375 - val_accuracy: 0.3262 - lr: 1.0000e-05
Epoch 34/60
75/75 [==============================] - 156s 2s/step - loss: 1.4607 - accuracy:
0.5365 - val_loss: 2.5797 - val_accuracy: 0.3497 - lr: 1.0000e-05
Epoch 35/60
75/75 [==============================] - 156s 2s/step - loss: 1.4665 - accuracy:
0.5349 - val_loss: 2.6038 - val_accuracy: 0.3354 - lr: 1.0000e-05
```

```
Epoch 36/60
75/75 [==============================] - 156s 2s/step - loss: 1.4161 - accuracy:
0.5491 - val_loss: 2.4891 - val_accuracy: 0.3497 - lr: 1.0000e-05
Epoch 37/60
75/75 [==============================] - 156s 2s/step - loss: 1.3521 - accuracy:
0.5618 - val_loss: 2.4806 - val_accuracy: 0.3409 - lr: 1.0000e-05
Epoch 38/60
75/75 [==============================] - 157s 2s/step - loss: 1.3572 - accuracy:
0.5594 - val_loss: 2.5073 - val_accuracy: 0.3522 - lr: 1.0000e-05
Epoch 39/60
75/75 [==============================] - 157s 2s/step - loss: 1.3190 - accuracy:
0.5749 - val_loss: 2.4844 - val_accuracy: 0.3661 - lr: 1.0000e-05
Epoch 40/60
75/75 [==============================] - 157s 2s/step - loss: 1.2797 - accuracy:
0.5804 - val_loss: 2.3630 - val_accuracy: 0.3623 - lr: 1.0000e-05
Epoch 41/60
75/75 [==============================] - 157s 2s/step - loss: 1.2880 - accuracy:
0.5716 - val_loss: 2.4211 - val_accuracy: 0.3678 - lr: 1.0000e-05
Epoch 42/60
75/75 [==============================] - 157s 2s/step - loss: 1.2534 - accuracy:
0.5813 - val_loss: 2.3468 - val_accuracy: 0.3770 - lr: 1.0000e-05
Epoch 43/60
75/75 [==============================] - 157s 2s/step - loss: 1.2436 - accuracy:
0.5883 - val_loss: 2.2621 - val_accuracy: 0.3834 - lr: 1.0000e-05
Epoch 44/60
75/75 [==============================] - 157s 2s/step - loss: 1.2113 - accuracy:
0.5912 - val_loss: 2.3245 - val_accuracy: 0.3783 - lr: 1.0000e-05
Epoch 45/60
75/75 [==============================] - 156s 2s/step - loss: 1.1845 - accuracy:
0.5961 - val_loss: 2.2710 - val_accuracy: 0.3947 - lr: 1.0000e-05
Epoch 46/60
75/75 [==============================] - 156s 2s/step - loss: 1.1710 - accuracy:
0.6142 - val_loss: 2.2570 - val_accuracy: 0.3859 - lr: 1.0000e-05
Epoch 47/60
75/75 [==============================] - 156s 2s/step - loss: 1.1693 - accuracy:
0.6014 - val_loss: 2.1918 - val_accuracy: 0.3859 - lr: 1.0000e-05
Epoch 48/60
75/75 [==============================] - 157s 2s/step - loss: 1.1113 - accuracy:
0.6183 - val_loss: 2.1769 - val_accuracy: 0.3964 - lr: 1.0000e-05
Epoch 49/60
75/75 [==============================] - 156s 2s/step - loss: 1.1135 - accuracy:
0.6178 - val_loss: 2.2051 - val_accuracy: 0.3901 - lr: 1.0000e-05
Epoch 50/60
75/75 [==============================] - 156s 2s/step - loss: 1.0892 - accuracy:
0.6253 - val_loss: 2.1784 - val_accuracy: 0.4044 - lr: 1.0000e-05
Epoch 51/60
75/75 [==============================] - 156s 2s/step - loss: 1.0570 - accuracy:
0.6284 - val_loss: 2.1371 - val_accuracy: 0.3968 - lr: 1.0000e-05
```

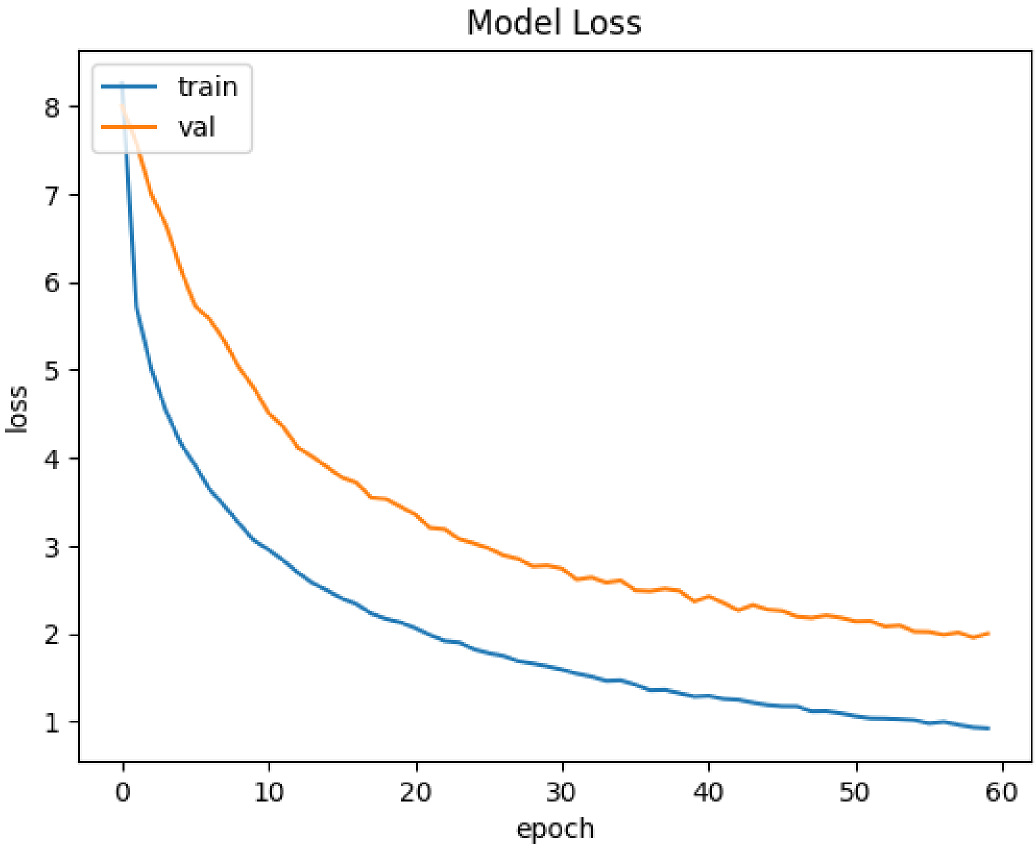
```
Epoch 52/60
75/75 [==============================] - 156s 2s/step - loss: 1.0336 - accuracy:
0.6495 - val_loss: 2.1407 - val_accuracy: 0.4124 - lr: 1.0000e-05
Epoch 53/60
75/75 [==============================] - 156s 2s/step - loss: 1.0309 - accuracy:
0.6420 - val_loss: 2.0781 - val_accuracy: 0.4140 - lr: 1.0000e-05
Epoch 54/60
75/75 [==============================] - 156s 2s/step - loss: 1.0228 - accuracy:
0.6468 - val_loss: 2.0908 - val_accuracy: 0.4111 - lr: 1.0000e-05
Epoch 55/60
75/75 [==============================] - 156s 2s/step - loss: 1.0120 - accuracy:
0.6450 - val_loss: 2.0211 - val_accuracy: 0.4237 - lr: 1.0000e-05
Epoch 56/60
75/75 [==============================] - 156s 2s/step - loss: 0.9756 - accuracy:
0.6550 - val_loss: 2.0174 - val_accuracy: 0.4258 - lr: 1.0000e-05
Epoch 57/60
75/75 [==============================] - 157s 2s/step - loss: 0.9928 - accuracy:
0.6451 - val_loss: 1.9832 - val_accuracy: 0.4208 - lr: 1.0000e-05
Epoch 58/60
75/75 [==============================] - 156s 2s/step - loss: 0.9586 - accuracy:
0.6613 - val_loss: 2.0106 - val_accuracy: 0.4216 - lr: 1.0000e-05
Epoch 59/60
75/75 [==============================] - 156s 2s/step - loss: 0.9300 - accuracy:
0.6639 - val_loss: 1.9527 - val_accuracy: 0.4296 - lr: 1.0000e-05
Epoch 60/60
75/75 [==============================] - 156s 2s/step - loss: 0.9190 - accuracy:
0.6678 - val_loss: 1.9975 - val_accuracy: 0.4279 - lr: 1.0000e-05
```

[ ]:

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



```
[ ]: plt.plot(history.history['loss'])
     plt.plot(history.history['val_loss'])
     plt.title('Model Loss')
     plt.ylabel('loss')
     plt.xlabel('epoch')
     plt.legend(['train', 'val'], loc='upper left')
     plt.show()
```



[ ]:

```
mobilev2model.save(os.path.
  ↪join(SAVE_DIR,'mobileNetV2_PretrainImageNet_128BatchSize_DataAug.h5'))
```

1.12 Train MobileNetV3 with Filtered Dataset and Data Augmentation, 128 Batch Size, Pretrained on ImageNet

Based on the Keras Documentation, `MobileNetV3Large` was chosen as the version of MobileNetV3 to use, as pruning/quantization will be later applied.

[ ]:

```
from tensorflow.keras import regularizers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define data augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

test_datagen = ImageDataGenerator(rescale=1./255)

# Define regularization
l2_reg = 0.001

# Load MobileNetV3Large model without top layer
mobilev3model = tf.keras.applications.MobileNetV3Large(
    input_shape=(224,224,3),
    alpha=1.0,
    include_top=False,
    weights='imagenet',
    pooling='max'
)

# Add a new top layer to match number of classes
x = mobilev3model.output
x = tf.keras.layers.Dense(128, activation='relu',␣
 ↪kernel_regularizer=regularizers.l2(l2_reg))(x)
output = tf.keras.layers.Dense(num_classes, activation='softmax',␣
 ↪kernel_regularizer=regularizers.l2(l2_reg))(x)

# Create new model with the base model and new top layers
mobilev3model = tf.keras.models.Model(inputs=mobilev3model.input,␣
 ↪outputs=output)

mobilev3model.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/mobilenet_v3/weights_mobilenet_v3_large_224_1.0_float_no_top_v2.h5
12683000/12683000 [==============================] - 1s 0us/step
Model: "model_1"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_4 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 rescaling_1 (Rescaling)        (None, 224, 224, 3)  0
['input_4[0][0]']

 Conv (Conv2D)                  (None, 112, 112, 16  432
['rescaling_1[0][0]']
                                )
 Conv/BatchNorm (BatchNormaliza  (None, 112, 112, 16  64          ['Conv[0][0]']
 tion)                          )

 tf.__operators__.add_29 (TFOpL  (None, 112, 112, 16  0
['Conv/BatchNorm[0][0]']
 ambda)                         )

 re_lu_40 (ReLU)                (None, 112, 112, 16  0
['tf.__operators__.add_29[0][0]']
                                )

 tf.math.multiply_29 (TFOpLambd  (None, 112, 112, 16  0
['re_lu_40[0][0]']
 a)                             )

 multiply_21 (Multiply)         (None, 112, 112, 16  0
['Conv/BatchNorm[0][0]',
                                )
'tf.math.multiply_29[0][0]']

 expanded_conv/depthwise (Depth  (None, 112, 112, 16  144
['multiply_21[0][0]']
 wiseConv2D)                    )

 expanded_conv/depthwise/BatchN  (None, 112, 112, 16  64
['expanded_conv/depthwise[0][0]']
 orm (BatchNormalization)       )

 re_lu_41 (ReLU)                (None, 112, 112, 16  0
['expanded_conv/depthwise/BatchNo
                                )                                 rm[0][0]']

 expanded_conv/project (Conv2D)  (None, 112, 112, 16  256
['re_lu_41[0][0]']
                                )

 expanded_conv/project/BatchNor  (None, 112, 112, 16  64
['expanded_conv/project[0][0]']
 m (BatchNormalization)         )

 expanded_conv/Add (Add)        (None, 112, 112, 16  0
['multiply_21[0][0]',
                                )
'expanded_conv/project/BatchNorm
                                                                  [0][0]']

 expanded_conv_1/expand (Conv2D  (None, 112, 112, 64  1024
['expanded_conv/Add[0][0]']
 )                              )

 expanded_conv_1/expand/BatchNo  (None, 112, 112, 64  256
['expanded_conv_1/expand[0][0]']
 rm (BatchNormalization)        )

 re_lu_42 (ReLU)                (None, 112, 112, 64  0
['expanded_conv_1/expand/BatchNor
                                )                               m[0][0]']

 expanded_conv_1/depthwise/pad   (None, 113, 113, 64  0
['re_lu_42[0][0]']
 (ZeroPadding2D)                )

 expanded_conv_1/depthwise (Dep  (None, 56, 56, 64)  576
['expanded_conv_1/depthwise/pad[0
 thwiseConv2D)                                                  ][0]']

 expanded_conv_1/depthwise/Batc  (None, 56, 56, 64)  256
['expanded_conv_1/depthwise[0][0]
 hNorm (BatchNormalization)                                     ']

 re_lu_43 (ReLU)                (None, 56, 56, 64)   0
['expanded_conv_1/depthwise/Batch
                                                                Norm[0][0]']

 expanded_conv_1/project (Conv2  (None, 56, 56, 24)  1536
['re_lu_43[0][0]']
 D)

 expanded_conv_1/project/BatchN  (None, 56, 56, 24)  96
['expanded_conv_1/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_2/expand (Conv2D  (None, 56, 56, 72)  1728
['expanded_conv_1/project/BatchNo
 )                                                              rm[0][0]']

 expanded_conv_2/expand/BatchNo  (None, 56, 56, 72)  288
['expanded_conv_2/expand[0][0]']
 rm (BatchNormalization)

 re_lu_44 (ReLU)                (None, 56, 56, 72)   0
['expanded_conv_2/expand/BatchNor
                                                                m[0][0]']

 expanded_conv_2/depthwise (Dep  (None, 56, 56, 72)  648
['re_lu_44[0][0]']
 thwiseConv2D)

 expanded_conv_2/depthwise/Batc  (None, 56, 56, 72)  288
['expanded_conv_2/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_45 (ReLU)                (None, 56, 56, 72)   0
['expanded_conv_2/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_2/project (Conv2  (None, 56, 56, 24)  1728
['re_lu_45[0][0]']
 D)

 expanded_conv_2/project/BatchN  (None, 56, 56, 24)  96
['expanded_conv_2/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_2/Add (Add)      (None, 56, 56, 24)   0
['expanded_conv_1/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_2/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_3/expand (Conv2D  (None, 56, 56, 72)  1728
['expanded_conv_2/Add[0][0]']
 )

 expanded_conv_3/expand/BatchNo  (None, 56, 56, 72)  288
['expanded_conv_3/expand[0][0]']
 rm (BatchNormalization)

 re_lu_46 (ReLU)                (None, 56, 56, 72)   0
['expanded_conv_3/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_3/depthwise/pad   (None, 59, 59, 72)  0
['re_lu_46[0][0]']
 (ZeroPadding2D)

 expanded_conv_3/depthwise (Dep  (None, 28, 28, 72)  1800
['expanded_conv_3/depthwise/pad[0
 thwiseConv2D)                                                   ][0]']

 expanded_conv_3/depthwise/Batc  (None, 28, 28, 72)  288
['expanded_conv_3/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']
 re_lu_47 (ReLU)                (None, 28, 28, 72)   0
['expanded_conv_3/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 72)    0
['re_lu_47[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_3/squeeze_excite  (None, 1, 1, 24)    1752
['expanded_conv_3/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 24)    0
['expanded_conv_3/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 72)    1800
['expanded_conv_3/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']

 tf.__operators__.add_30 (TFOpL  (None, 1, 1, 72)    0
['expanded_conv_3/squeeze_excite/
 ambda)                                                          Conv_1[0][0]']

 re_lu_48 (ReLU)                (None, 1, 1, 72)     0
['tf.__operators__.add_30[0][0]']

 tf.math.multiply_30 (TFOpLambd  (None, 1, 1, 72)    0
['re_lu_48[0][0]']
 a)

 expanded_conv_3/squeeze_excite  (None, 28, 28, 72)  0
['re_lu_47[0][0]',
 /Mul (Multiply)
'tf.math.multiply_30[0][0]']

 expanded_conv_3/project (Conv2  (None, 28, 28, 40)  2880
['expanded_conv_3/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_3/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_3/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_4/expand (Conv2D  (None, 28, 28, 120)  4800
['expanded_conv_3/project/BatchNo
```

```
 )                                                                rm[0][0]']

 expanded_conv_4/expand/BatchNo  (None, 28, 28, 120)  480
['expanded_conv_4/expand[0][0]']
 rm (BatchNormalization)

 re_lu_49 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_4/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_4/depthwise (Dep  (None, 28, 28, 120)  3000
['re_lu_49[0][0]']
 thwiseConv2D)

 expanded_conv_4/depthwise/Batc  (None, 28, 28, 120)  480
['expanded_conv_4/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_50 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_4/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 120)   0
['re_lu_50[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_4/squeeze_excite  (None, 1, 1, 32)    3872
['expanded_conv_4/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 32)    0
['expanded_conv_4/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 120)   3960
['expanded_conv_4/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']

 tf.__operators__.add_31 (TFOpL  (None, 1, 1, 120)   0
['expanded_conv_4/squeeze_excite/
 ambda)                                                          Conv_1[0][0]']

 re_lu_51 (ReLU)                (None, 1, 1, 120)    0
['tf.__operators__.add_31[0][0]']

 tf.math.multiply_31 (TFOpLambd  (None, 1, 1, 120)   0
['re_lu_51[0][0]']
```

```
 a)

 expanded_conv_4/squeeze_excite  (None, 28, 28, 120)  0
['re_lu_50[0][0]',
 /Mul (Multiply)
'tf.math.multiply_31[0][0]']

 expanded_conv_4/project (Conv2  (None, 28, 28, 40)  4800
['expanded_conv_4/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_4/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_4/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_4/Add (Add)      (None, 28, 28, 40)   0
['expanded_conv_3/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_4/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_5/expand (Conv2D  (None, 28, 28, 120)  4800
['expanded_conv_4/Add[0][0]']
 )

 expanded_conv_5/expand/BatchNo  (None, 28, 28, 120)  480
['expanded_conv_5/expand[0][0]']
 rm (BatchNormalization)

 re_lu_52 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_5/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_5/depthwise (Dep  (None, 28, 28, 120)  3000
['re_lu_52[0][0]']
 thwiseConv2D)

 expanded_conv_5/depthwise/Batc  (None, 28, 28, 120)  480
['expanded_conv_5/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_53 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_5/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 120)   0
['re_lu_53[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_5/squeeze_excite  (None, 1, 1, 32)    3872
['expanded_conv_5/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 32)    0
['expanded_conv_5/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 120)   3960
['expanded_conv_5/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']

 tf.__operators__.add_32 (TFOpL  (None, 1, 1, 120)   0
['expanded_conv_5/squeeze_excite/
 ambda)                                                          Conv_1[0][0]']

 re_lu_54 (ReLU)                (None, 1, 1, 120)    0
['tf.__operators__.add_32[0][0]']

 tf.math.multiply_32 (TFOpLambd  (None, 1, 1, 120)   0
['re_lu_54[0][0]']
 a)

 expanded_conv_5/squeeze_excite  (None, 28, 28, 120)  0
['re_lu_53[0][0]',
 /Mul (Multiply)
'tf.math.multiply_32[0][0]']

 expanded_conv_5/project (Conv2  (None, 28, 28, 40)  4800
['expanded_conv_5/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_5/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_5/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_5/Add (Add)      (None, 28, 28, 40)   0
['expanded_conv_4/Add[0][0]',
'expanded_conv_5/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_6/expand (Conv2D  (None, 28, 28, 240)  9600
['expanded_conv_5/Add[0][0]']
 )

 expanded_conv_6/expand/BatchNo  (None, 28, 28, 240)  960
```

```
['expanded_conv_6/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_33 (TFOpL  (None, 28, 28, 240)  0
['expanded_conv_6/expand/BatchNor
 ambda)                                                           m[0][0]']

 re_lu_55 (ReLU)                (None, 28, 28, 240)  0
['tf.__operators__.add_33[0][0]']

 tf.math.multiply_33 (TFOpLambd  (None, 28, 28, 240)  0
['re_lu_55[0][0]']
 a)

 multiply_22 (Multiply)         (None, 28, 28, 240)  0
['expanded_conv_6/expand/BatchNor
                                                                  m[0][0]',
'tf.math.multiply_33[0][0]']

 expanded_conv_6/depthwise/pad   (None, 29, 29, 240)  0
['multiply_22[0][0]']
 (ZeroPadding2D)

 expanded_conv_6/depthwise (Dep  (None, 14, 14, 240)  2160
['expanded_conv_6/depthwise/pad[0
 thwiseConv2D)                                                    ][0]']

 expanded_conv_6/depthwise/Batc  (None, 14, 14, 240)  960
['expanded_conv_6/depthwise[0][0]
 hNorm (BatchNormalization)                                       ']

 tf.__operators__.add_34 (TFOpL  (None, 14, 14, 240)  0
['expanded_conv_6/depthwise/Batch
 ambda)                                                           Norm[0][0]']

 re_lu_56 (ReLU)                (None, 14, 14, 240)  0
['tf.__operators__.add_34[0][0]']

 tf.math.multiply_34 (TFOpLambd  (None, 14, 14, 240)  0
['re_lu_56[0][0]']
 a)

 multiply_23 (Multiply)         (None, 14, 14, 240)  0
['expanded_conv_6/depthwise/Batch
                                                                  Norm[0][0]',
'tf.math.multiply_34[0][0]']

 expanded_conv_6/project (Conv2  (None, 14, 14, 80)  19200
['multiply_23[0][0]']
 D)

 expanded_conv_6/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_6/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_7/expand (Conv2D  (None, 14, 14, 200)  16000
['expanded_conv_6/project/BatchNo
 )                                                               rm[0][0]']

 expanded_conv_7/expand/BatchNo  (None, 14, 14, 200)  800
['expanded_conv_7/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_35 (TFOpL  (None, 14, 14, 200)  0
['expanded_conv_7/expand/BatchNor
 ambda)                                                          m[0][0]']

 re_lu_57 (ReLU)                (None, 14, 14, 200)  0
['tf.__operators__.add_35[0][0]']

 tf.math.multiply_35 (TFOpLambd  (None, 14, 14, 200)  0
['re_lu_57[0][0]']
 a)

 multiply_24 (Multiply)         (None, 14, 14, 200)  0
['expanded_conv_7/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_35[0][0]']

 expanded_conv_7/depthwise (Dep  (None, 14, 14, 200)  1800
['multiply_24[0][0]']
 thwiseConv2D)

 expanded_conv_7/depthwise/Batc  (None, 14, 14, 200)  800
['expanded_conv_7/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 tf.__operators__.add_36 (TFOpL  (None, 14, 14, 200)  0
['expanded_conv_7/depthwise/Batch
 ambda)                                                          Norm[0][0]']

 re_lu_58 (ReLU)                (None, 14, 14, 200)  0
['tf.__operators__.add_36[0][0]']

 tf.math.multiply_36 (TFOpLambd  (None, 14, 14, 200)  0
['re_lu_58[0][0]']
```

```
 a)

 multiply_25 (Multiply)         (None, 14, 14, 200)  0
['expanded_conv_7/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_36[0][0]']

 expanded_conv_7/project (Conv2  (None, 14, 14, 80)  16000
['multiply_25[0][0]']
 D)

 expanded_conv_7/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_7/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_7/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_6/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_7/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_8/expand (Conv2D  (None, 14, 14, 184)  14720
['expanded_conv_7/Add[0][0]']
 )

 expanded_conv_8/expand/BatchNo  (None, 14, 14, 184)  736
['expanded_conv_8/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_37 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_8/expand/BatchNor
 ambda)                                                          m[0][0]']

 re_lu_59 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_37[0][0]']

 tf.math.multiply_37 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_59[0][0]']
 a)

 multiply_26 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_8/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_37[0][0]']

 expanded_conv_8/depthwise (Dep  (None, 14, 14, 184)  1656
['multiply_26[0][0]']
 thwiseConv2D)
 expanded_conv_8/depthwise/Batc  (None, 14, 14, 184)  736
['expanded_conv_8/depthwise[0][0]
 hNorm (BatchNormalization)                                       ']

 tf.__operators__.add_38 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_8/depthwise/Batch
 ambda)                                                          Norm[0][0]']

 re_lu_60 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_38[0][0]']

 tf.math.multiply_38 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_60[0][0]']
 a)

 multiply_27 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_8/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_38[0][0]']

 expanded_conv_8/project (Conv2  (None, 14, 14, 80)  14720
['multiply_27[0][0]']
 D)

 expanded_conv_8/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_8/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_8/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_7/Add[0][0]',
'expanded_conv_8/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_9/expand (Conv2D  (None, 14, 14, 184)  14720
['expanded_conv_8/Add[0][0]']
 )

 expanded_conv_9/expand/BatchNo  (None, 14, 14, 184)  736
['expanded_conv_9/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_39 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_9/expand/BatchNor
 ambda)                                                          m[0][0]']

 re_lu_61 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_39[0][0]']
```

```
 tf.math.multiply_39 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_61[0][0]']
 a)

 multiply_28 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_9/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_39[0][0]']

 expanded_conv_9/depthwise (Dep  (None, 14, 14, 184)  1656
['multiply_28[0][0]']
 thwiseConv2D)

 expanded_conv_9/depthwise/Batc  (None, 14, 14, 184)  736
['expanded_conv_9/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 tf.__operators__.add_40 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_9/depthwise/Batch
 ambda)                                                          Norm[0][0]']

 re_lu_62 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_40[0][0]']

 tf.math.multiply_40 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_62[0][0]']
 a)

 multiply_29 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_9/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_40[0][0]']

 expanded_conv_9/project (Conv2  (None, 14, 14, 80)   14720
['multiply_29[0][0]']
 D)

 expanded_conv_9/project/BatchN  (None, 14, 14, 80)   320
['expanded_conv_9/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_9/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_8/Add[0][0]',
'expanded_conv_9/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_10/expand (Conv2  (None, 14, 14, 480)  38400
['expanded_conv_9/Add[0][0]']
 D)

 expanded_conv_10/expand/BatchN  (None, 14, 14, 480)  1920
['expanded_conv_10/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_41 (TFOpL  (None, 14, 14, 480)  0
['expanded_conv_10/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_63 (ReLU)                (None, 14, 14, 480)  0
['tf.__operators__.add_41[0][0]']

 tf.math.multiply_41 (TFOpLambd  (None, 14, 14, 480)  0
['re_lu_63[0][0]']
 a)

 multiply_30 (Multiply)         (None, 14, 14, 480)  0
['expanded_conv_10/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_41[0][0]']

 expanded_conv_10/depthwise (De  (None, 14, 14, 480)  4320
['multiply_30[0][0]']
 pthwiseConv2D)

 expanded_conv_10/depthwise/Bat  (None, 14, 14, 480)  1920
['expanded_conv_10/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_42 (TFOpL  (None, 14, 14, 480)  0
['expanded_conv_10/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_64 (ReLU)                (None, 14, 14, 480)  0
['tf.__operators__.add_42[0][0]']

 tf.math.multiply_42 (TFOpLambd  (None, 14, 14, 480)  0
['re_lu_64[0][0]']
 a)

 multiply_31 (Multiply)         (None, 14, 14, 480)  0
['expanded_conv_10/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_42[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 480)   0
```

```
['multiply_31[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_10/squeeze_excit  (None, 1, 1, 120)   57720
['expanded_conv_10/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 120)   0
['expanded_conv_10/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 480)   58080
['expanded_conv_10/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_43 (TFOpL  (None, 1, 1, 480)   0
['expanded_conv_10/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_65 (ReLU)                (None, 1, 1, 480)    0
['tf.__operators__.add_43[0][0]']

 tf.math.multiply_43 (TFOpLambd  (None, 1, 1, 480)   0
['re_lu_65[0][0]']
 a)

 expanded_conv_10/squeeze_excit  (None, 14, 14, 480)  0
['multiply_31[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_43[0][0]']

 expanded_conv_10/project (Conv  (None, 14, 14, 112)  53760
['expanded_conv_10/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_10/project/Batch  (None, 14, 14, 112)  448
['expanded_conv_10/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_11/expand (Conv2  (None, 14, 14, 672)  75264
['expanded_conv_10/project/BatchN
 D)                                                              orm[0][0]']

 expanded_conv_11/expand/BatchN  (None, 14, 14, 672)  2688
['expanded_conv_11/expand[0][0]']
 orm (BatchNormalization)
 tf.__operators__.add_44 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_11/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_66 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_44[0][0]']

 tf.math.multiply_44 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_66[0][0]']
 a)

 multiply_32 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_11/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_44[0][0]']

 expanded_conv_11/depthwise (De  (None, 14, 14, 672)  6048
['multiply_32[0][0]']
 pthwiseConv2D)

 expanded_conv_11/depthwise/Bat  (None, 14, 14, 672)  2688
['expanded_conv_11/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_45 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_11/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_67 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_45[0][0]']

 tf.math.multiply_45 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_67[0][0]']
 a)

 multiply_33 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_11/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_45[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 672)   0
['multiply_33[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_11/squeeze_excit  (None, 1, 1, 168)   113064
['expanded_conv_11/squeeze_excite
```

```
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 168)   0
['expanded_conv_11/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 672)   113568
['expanded_conv_11/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_46 (TFOpL  (None, 1, 1, 672)   0
['expanded_conv_11/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_68 (ReLU)                (None, 1, 1, 672)    0
['tf.__operators__.add_46[0][0]']

 tf.math.multiply_46 (TFOpLambd  (None, 1, 1, 672)   0
['re_lu_68[0][0]']
 a)

 expanded_conv_11/squeeze_excit  (None, 14, 14, 672)  0
['multiply_33[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_46[0][0]']

 expanded_conv_11/project (Conv  (None, 14, 14, 112)  75264
['expanded_conv_11/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_11/project/Batch  (None, 14, 14, 112)  448
['expanded_conv_11/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_11/Add (Add)     (None, 14, 14, 112)  0
['expanded_conv_10/project/BatchN
                                                                 orm[0][0]',
'expanded_conv_11/project/BatchN
                                                                 orm[0][0]']

 expanded_conv_12/expand (Conv2  (None, 14, 14, 672)  75264
['expanded_conv_11/Add[0][0]']
 D)

 expanded_conv_12/expand/BatchN  (None, 14, 14, 672)  2688
['expanded_conv_12/expand[0][0]']
 orm (BatchNormalization)
 tf.__operators__.add_47 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_12/expand/BatchNo
 ambda)                                                         rm[0][0]']

 re_lu_69 (ReLU)               (None, 14, 14, 672)  0
['tf.__operators__.add_47[0][0]']

 tf.math.multiply_47 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_69[0][0]']
 a)

 multiply_34 (Multiply)        (None, 14, 14, 672)  0
['expanded_conv_12/expand/BatchNo
                                                                rm[0][0]',
'tf.math.multiply_47[0][0]']

 expanded_conv_12/depthwise/pad  (None, 17, 17, 672)  0
['multiply_34[0][0]']
  (ZeroPadding2D)

 expanded_conv_12/depthwise (De  (None, 7, 7, 672)   16800
['expanded_conv_12/depthwise/pad[
 pthwiseConv2D)                                                 0][0]']

 expanded_conv_12/depthwise/Bat  (None, 7, 7, 672)   2688
['expanded_conv_12/depthwise[0][0
 chNorm (BatchNormalization)                                    ]']

 tf.__operators__.add_48 (TFOpL  (None, 7, 7, 672)   0
['expanded_conv_12/depthwise/Batc
 ambda)                                                         hNorm[0][0]']

 re_lu_70 (ReLU)               (None, 7, 7, 672)    0
['tf.__operators__.add_48[0][0]']

 tf.math.multiply_48 (TFOpLambd  (None, 7, 7, 672)   0
['re_lu_70[0][0]']
 a)

 multiply_35 (Multiply)        (None, 7, 7, 672)    0
['expanded_conv_12/depthwise/Batc
                                                                hNorm[0][0]',
'tf.math.multiply_48[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 672)   0
['multiply_35[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_12/squeeze_excit  (None, 1, 1, 168)   113064
['expanded_conv_12/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 168)   0
['expanded_conv_12/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 672)   113568
['expanded_conv_12/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_49 (TFOpL  (None, 1, 1, 672)   0
['expanded_conv_12/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_71 (ReLU)                (None, 1, 1, 672)    0
['tf.__operators__.add_49[0][0]']

 tf.math.multiply_49 (TFOpLambd  (None, 1, 1, 672)   0
['re_lu_71[0][0]']
 a)

 expanded_conv_12/squeeze_excit  (None, 7, 7, 672)   0
['multiply_35[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_49[0][0]']

 expanded_conv_12/project (Conv  (None, 7, 7, 160)   107520
['expanded_conv_12/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_12/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_12/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_13/expand (Conv2  (None, 7, 7, 960)   153600
['expanded_conv_12/project/BatchN
 D)                                                              orm[0][0]']

 expanded_conv_13/expand/BatchN  (None, 7, 7, 960)   3840
['expanded_conv_13/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_50 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_13/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_72 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_50[0][0]']

 tf.math.multiply_50 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_72[0][0]']
 a)

 multiply_36 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_13/expand/BatchNo
                                                                 rm[0][0]',

'tf.math.multiply_50[0][0]']

 expanded_conv_13/depthwise (De  (None, 7, 7, 960)   24000
['multiply_36[0][0]']
 pthwiseConv2D)

 expanded_conv_13/depthwise/Bat  (None, 7, 7, 960)   3840
['expanded_conv_13/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_51 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_13/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_73 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_51[0][0]']

 tf.math.multiply_51 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_73[0][0]']
 a)

 multiply_37 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_13/depthwise/Batc
                                                                 hNorm[0][0]',

'tf.math.multiply_51[0][0]']

 expanded_conv_13/squeeze_excit  (None, 1, 1, 960)   0
['multiply_37[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_13/squeeze_excit  (None, 1, 1, 240)   230640
['expanded_conv_13/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']
```

```
 expanded_conv_13/squeeze_excit  (None, 1, 1, 240)   0
['expanded_conv_13/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_13/squeeze_excit  (None, 1, 1, 960)   231360
['expanded_conv_13/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_52 (TFOpL  (None, 1, 1, 960)   0
['expanded_conv_13/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_74 (ReLU)                (None, 1, 1, 960)    0
['tf.__operators__.add_52[0][0]']

 tf.math.multiply_52 (TFOpLambd  (None, 1, 1, 960)   0
['re_lu_74[0][0]']
 a)

 expanded_conv_13/squeeze_excit  (None, 7, 7, 960)   0
['multiply_37[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_52[0][0]']

 expanded_conv_13/project (Conv  (None, 7, 7, 160)   153600
['expanded_conv_13/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_13/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_13/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_13/Add (Add)     (None, 7, 7, 160)    0
['expanded_conv_12/project/BatchN
                                                                 orm[0][0]',
'expanded_conv_13/project/BatchN
                                                                 orm[0][0]']

 expanded_conv_14/expand (Conv2  (None, 7, 7, 960)   153600
['expanded_conv_13/Add[0][0]']
 D)

 expanded_conv_14/expand/BatchN  (None, 7, 7, 960)   3840
['expanded_conv_14/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_53 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_14/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_75 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_53[0][0]']

 tf.math.multiply_53 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_75[0][0]']
 a)

 multiply_38 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_14/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_53[0][0]']

 expanded_conv_14/depthwise (De  (None, 7, 7, 960)   24000
['multiply_38[0][0]']
 pthwiseConv2D)

 expanded_conv_14/depthwise/Bat  (None, 7, 7, 960)   3840
['expanded_conv_14/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_54 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_14/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_76 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_54[0][0]']

 tf.math.multiply_54 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_76[0][0]']
 a)

 multiply_39 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_14/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_54[0][0]']

 expanded_conv_14/squeeze_excit  (None, 1, 1, 960)   0
['multiply_39[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_14/squeeze_excit  (None, 1, 1, 240)   230640
['expanded_conv_14/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']
```

```
 expanded_conv_14/squeeze_excit  (None, 1, 1, 240)   0
['expanded_conv_14/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_14/squeeze_excit  (None, 1, 1, 960)   231360
['expanded_conv_14/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_55 (TFOpL  (None, 1, 1, 960)   0
['expanded_conv_14/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_77 (ReLU)                (None, 1, 1, 960)    0
['tf.__operators__.add_55[0][0]']

 tf.math.multiply_55 (TFOpLambd  (None, 1, 1, 960)   0
['re_lu_77[0][0]']
 a)

 expanded_conv_14/squeeze_excit  (None, 7, 7, 960)   0
['multiply_39[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_55[0][0]']

 expanded_conv_14/project (Conv  (None, 7, 7, 160)   153600
['expanded_conv_14/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_14/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_14/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_14/Add (Add)     (None, 7, 7, 160)    0
['expanded_conv_13/Add[0][0]',
'expanded_conv_14/project/BatchN
                                                                 orm[0][0]']

 Conv_1 (Conv2D)                (None, 7, 7, 960)    153600
['expanded_conv_14/Add[0][0]']

 Conv_1/BatchNorm (BatchNormali  (None, 7, 7, 960)   3840
['Conv_1[0][0]']
 zation)

 tf.__operators__.add_56 (TFOpL  (None, 7, 7, 960)   0
['Conv_1/BatchNorm[0][0]']
 ambda)
 re_lu_78 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_56[0][0]']

 tf.math.multiply_56 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_78[0][0]']
 a)

 multiply_40 (Multiply)         (None, 7, 7, 960)    0
['Conv_1/BatchNorm[0][0]',
'tf.math.multiply_56[0][0]']

 max_pool (GlobalMaxPooling2D)  (None, 960)          0
['multiply_40[0][0]']

 dense_1 (Dense)                (None, 128)          123008
['max_pool[0][0]']

 dense_2 (Dense)                (None, 25)           3225
['dense_1[0][0]']

================================================================================
==================
Total params: 3,122,585
Trainable params: 3,098,185
Non-trainable params: 24,400
--------------------------------------------------------------------------------
------------------
```

[ ]:

```
# Call back 1:
base_learning_rate = 1e-5
opt1 = tf.keras.optimizers.Adam(learning_rate=base_learning_rate)
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='val_accuracy', min_delta=0, patience=4, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

# Call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1, callback_2]
```

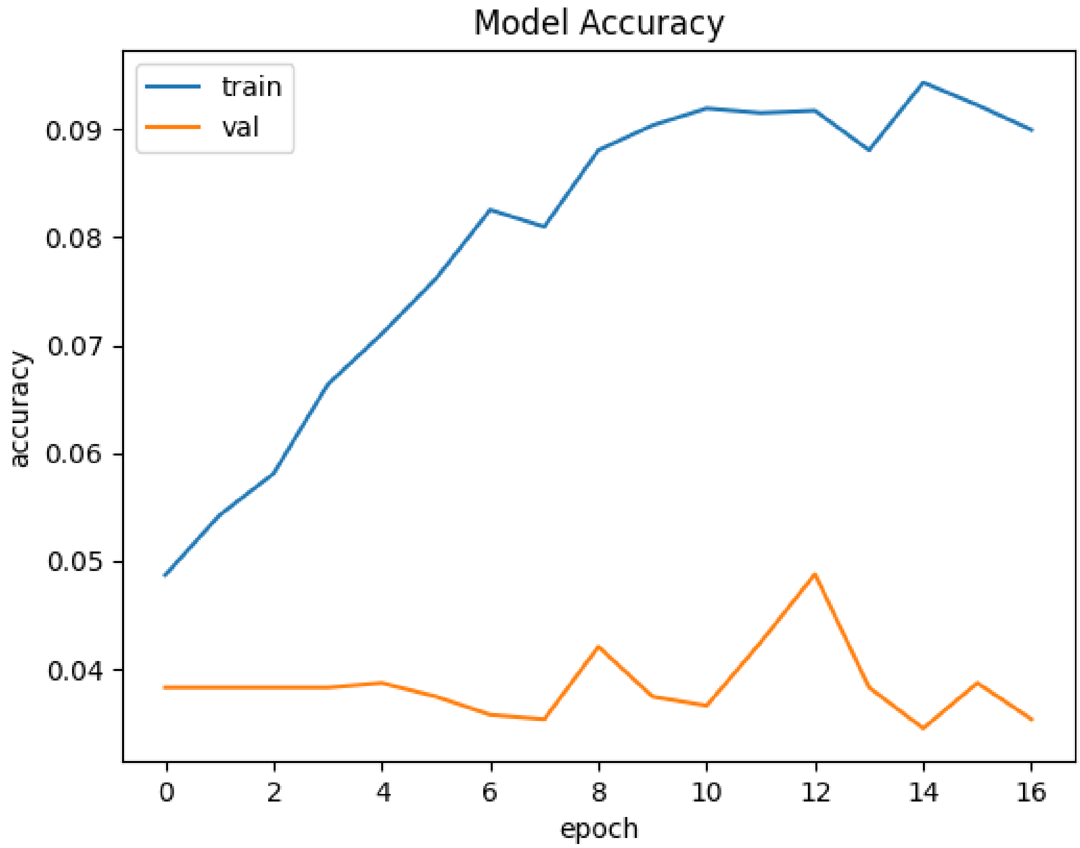
```python
#compiling our Model for dataset
mobilev3model.compile(optimizer=opt1,
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = mobilev3model.fit(
    train_generator,
    epochs=60,
    validation_data=validation_generator,
    callbacks=callback_list)
```

```
Epoch 1/60
75/75 [==============================] - 191s 2s/step - loss: 17.8301 -
accuracy: 0.0487 - val_loss: 5.0050 - val_accuracy: 0.0383 - lr: 1.0000e-05
Epoch 2/60
75/75 [==============================] - 156s 2s/step - loss: 12.9007 -
accuracy: 0.0542 - val_loss: 4.9525 - val_accuracy: 0.0383 - lr: 1.0000e-05
Epoch 3/60
75/75 [==============================] - 156s 2s/step - loss: 10.5733 -
accuracy: 0.0581 - val_loss: 4.8971 - val_accuracy: 0.0383 - lr: 1.0000e-05
Epoch 4/60
75/75 [==============================] - 156s 2s/step - loss: 9.2136 - accuracy:
0.0663 - val_loss: 4.8459 - val_accuracy: 0.0383 - lr: 1.0000e-05
Epoch 5/60
75/75 [==============================] - 156s 2s/step - loss: 8.2128 - accuracy:
0.0710 - val_loss: 4.7744 - val_accuracy: 0.0387 - lr: 1.0000e-05
Epoch 6/60
75/75 [==============================] - 156s 2s/step - loss: 7.2915 - accuracy:
0.0762 - val_loss: 4.7317 - val_accuracy: 0.0374 - lr: 1.0000e-05
Epoch 7/60
75/75 [==============================] - 156s 2s/step - loss: 6.5418 - accuracy:
0.0825 - val_loss: 4.7754 - val_accuracy: 0.0357 - lr: 1.0000e-05
Epoch 8/60
75/75 [==============================] - 156s 2s/step - loss: 6.0443 - accuracy:
0.0810 - val_loss: 5.1172 - val_accuracy: 0.0353 - lr: 1.0000e-05
Epoch 9/60
75/75 [==============================] - 156s 2s/step - loss: 5.5578 - accuracy:
0.0881 - val_loss: 5.5398 - val_accuracy: 0.0420 - lr: 1.0000e-05
Epoch 10/60
75/75 [==============================] - 156s 2s/step - loss: 5.0983 - accuracy:
0.0904 - val_loss: 6.3388 - val_accuracy: 0.0374 - lr: 1.0000e-05
Epoch 11/60
75/75 [==============================] - 156s 2s/step - loss: 4.8992 - accuracy:
0.0919 - val_loss: 6.5202 - val_accuracy: 0.0366 - lr: 1.0000e-06
Epoch 12/60
75/75 [==============================] - 156s 2s/step - loss: 4.8250 - accuracy:
```

```
0.0915 - val_loss: 6.3342 - val_accuracy: 0.0425 - lr: 1.0000e-06
Epoch 13/60
75/75 [==============================] - 156s 2s/step - loss: 4.8233 - accuracy:
0.0917 - val_loss: 6.1573 - val_accuracy: 0.0488 - lr: 1.0000e-06
Epoch 14/60
75/75 [==============================] - 157s 2s/step - loss: 4.7748 - accuracy:
0.0881 - val_loss: 6.0619 - val_accuracy: 0.0383 - lr: 1.0000e-06
Epoch 15/60
75/75 [==============================] - 156s 2s/step - loss: 4.7459 - accuracy:
0.0943 - val_loss: 6.0271 - val_accuracy: 0.0345 - lr: 1.0000e-07
Epoch 16/60
75/75 [==============================] - 156s 2s/step - loss: 4.7813 - accuracy:
0.0922 - val_loss: 6.1385 - val_accuracy: 0.0387 - lr: 1.0000e-07
Epoch 17/60
75/75 [==============================] - 156s 2s/step - loss: 4.7634 - accuracy:
0.0899 - val_loss: 6.2394 - val_accuracy: 0.0353 - lr: 1.0000e-07
```

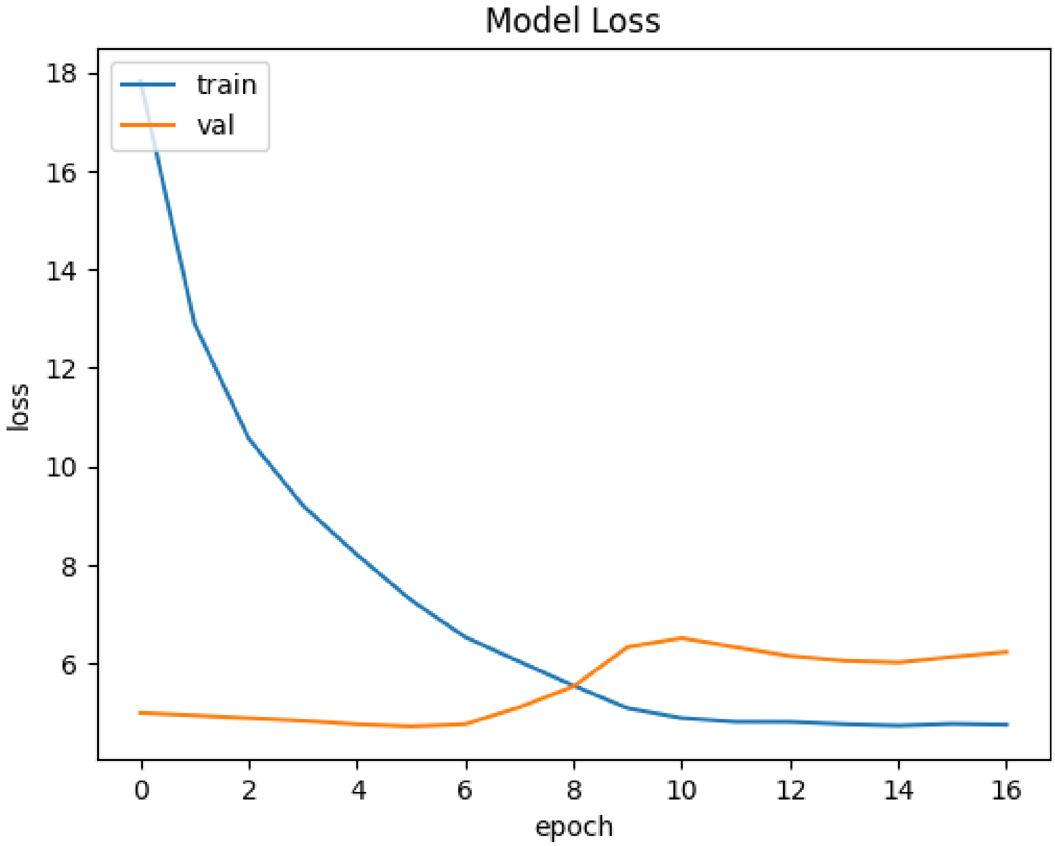
[ ]:

```python
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



[ ]:

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



[ ]:
```
mobilev3model.save(os.path.
  ↪join(SAVE_DIR,'mobileNetV3_PretrainImageNet_128BatchSize_DataAug.h5'))
```

1.13 Reducing the Filtered Data Set to Make/Model

As none of the models to date have been able to provide a good level of accuracy, this will be the final attempt to modify the data set to something that either MobileNetV2 or MobileNetV3 can perform accurate predictions after training. A suspicion is that though the data set was reduced substantially, there are still some classes that are very similar (

[ ]:
```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

[ ]:
```
import os
import shutil
from google.colab import drive
import tensorflow as tf
from matplotlib import pyplot as plt
```

```
DIRPATH = '/content/VMMRdb_LrgImgCount/'
SAVE_DIR = '/content/drive/MyDrive/endg511project/SavedModels/'

shutil.unpack_archive("/content/drive/MyDrive/endg511project/VMMRdb_LrgImgCount.
 ↪zip", DIRPATH)

num_classes = len(os.listdir(DIRPATH))
print(num_classes)
```

```
25
```

[ ]:

```
def make_dir(path_to_dir):
    if os.path.exists(path_to_dir):
        return path_to_dir
    else:
        os.mkdir(path_to_dir)
        return path_to_dir

NEW_DIR = make_dir('/content/VMMRdb_LrgImgCount_ReducedClasses/')

for i, dir in enumerate(os.listdir(DIRPATH)):
    new_file_dir = make_dir(NEW_DIR+"_".join(dir.split("_")[:2]))
    for j, file in enumerate(os.listdir(DIRPATH+'/'+dir)):
        old_file_loc = DIRPATH+dir+'/'+file
        new_file_loc = new_file_dir + '/' + file
        shutil.move(old_file_loc, new_file_loc)

# delete old directory of empty folders
shutil.rmtree(DIRPATH)
DIRPATH = NEW_DIR
num_classes = len(os.listdir(DIRPATH))
print(f"Number of Classes: {num_classes}")
for subdir in os.listdir(DIRPATH):
  print(f"Class: {subdir}, Count: {len(os.listdir(os.path.join(DIRPATH,␣
 ↪subdir)))}")
```

```
Number of Classes: 12
Class: honda_accord, Count: 2135
Class: nissan_altima, Count: 2439
Class: chevrolet_impala, Count: 436
Class: toyota_camry, Count: 467
Class: ford_mustang, Count: 504
Class: ford_explorer, Count: 1992
Class: ford_taurus, Count: 815
Class: volkswagen_jetta, Count: 402
Class: dodge_grand caravan, Count: 447
```

```
Class: honda_civic, Count: 1412
Class: ford_f150, Count: 414
Class: chevrolet_silverado, Count: 487
```

1.14 Training MobileNetV2 With Filtered and Reduced Data Set

[ ]:

```
DIRPATH = '/content/VMMRdb_LrgImgCount_ReducedClasses/'
```

[ ]:

```
train_mobilenet, test_mobilenet = tf.keras.utils.image_dataset_from_directory(
    DIRPATH,
    validation_split=0.2,
    subset="both",
    seed=123,
    image_size=[224,224],
    label_mode='int')
```

```
Found 11950 files belonging to 12 classes.
Using 9560 files for training.
Using 2390 files for validation.
```

Computing class weights

[ ]:

```
import numpy as np
from collections import Counter

# Get the class labels from the dataset
class_labels = train_mobilenet.class_names

# Get the number of images per class
num_images_per_class = []
for images, labels in train_mobilenet:
    for label in labels.numpy():
        num_images_per_class.append(label)
class_counts = Counter(num_images_per_class)

# Calculate class frequencies
total_num_images = sum(class_counts.values())
class_frequencies = []
for i in range(len(class_labels)):
    class_frequencies.append(class_counts[i] / total_num_images)

# Calculate class weights
max_frequency = max(class_frequencies)
class_weights = {}
for i in range(len(class_labels)):
    class_weights[i] = max_frequency / class_frequencies[i]

print(class_weights)
```

```
{0: 5.494318181818182, 1: 4.871536523929471, 2: 5.327823691460055, 3:
1.2217308907138344, 4: 5.7559523809523805, 5: 4.89620253164557, 6:
2.9753846153846153, 7: 1.1153402537485582, 8: 1.7267857142857141, 9: 1.0, 10:
5.212938005390836, 11: 5.950769230769231}
```

[ ]:
```python
mobilev2model = tf.keras.applications.MobileNetV2(
    input_shape=(224,224,3),
    alpha=1.0,
    include_top=True,
    weights=None,
    input_tensor=None,
    classes=num_classes,
    pooling='max'
)

mobilev2model.summary()
```

```
Model: "mobilenetv2_1.00_224"
--------------------------------------------------------------------------------
------------------
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_3 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864
['input_3[0][0]']
                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0][0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0
['bn_Conv1[0][0]']
                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288
['Conv1_relu[0][0]']
 wiseConv2D)                    )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128
['expanded_conv_depthwise[0][0]']
 tchNormalization)              )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0
['expanded_conv_depthwise_BN[0][0
 ReLU)                          )                                ]']
 expanded_conv_project (Conv2D)  (None, 112, 112, 16  512
['expanded_conv_depthwise_relu[0]
                                )                                [0]']

 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64
['expanded_conv_project[0][0]']
 hNormalization)                )

 block_1_expand (Conv2D)        (None, 112, 112, 96  1536
['expanded_conv_project_BN[0][0]'
                                )                                ]

 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384
['block_1_expand[0][0]']
 ization)                       )

 block_1_expand_relu (ReLU)     (None, 112, 112, 96  0
['block_1_expand_BN[0][0]']
                                )

 block_1_pad (ZeroPadding2D)    (None, 113, 113, 96  0
['block_1_expand_relu[0][0]']
                                )

 block_1_depthwise (DepthwiseCo  (None, 56, 56, 96)  864
['block_1_pad[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 56, 56, 96)  384
['block_1_depthwise[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 56, 56, 96)   0
['block_1_depthwise_BN[0][0]']

 block_1_project (Conv2D)       (None, 56, 56, 24)   2304
['block_1_depthwise_relu[0][0]']

 block_1_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_1_project[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_1_project_BN[0][0]']

 block_2_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_2_expand[0][0]']
```

```
 ization)

 block_2_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_2_expand_BN[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 56, 56, 144)  1296
['block_2_expand_relu[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 56, 56, 144)  576
['block_2_depthwise[0][0]']
 malization)

 block_2_depthwise_relu (ReLU)  (None, 56, 56, 144)  0
['block_2_depthwise_BN[0][0]']

 block_2_project (Conv2D)       (None, 56, 56, 24)   3456
['block_2_depthwise_relu[0][0]']

 block_2_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_2_project[0][0]']
 lization)

 block_2_add (Add)              (None, 56, 56, 24)   0
['block_1_project_BN[0][0]',
'block_2_project_BN[0][0]']

 block_3_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_2_add[0][0]']

 block_3_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_3_expand[0][0]']
 ization)

 block_3_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_3_expand_BN[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 57, 57, 144)  0
['block_3_expand_relu[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 28, 28, 144)  1296
['block_3_pad[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 28, 28, 144)  576
['block_3_depthwise[0][0]']
 malization)
 block_3_depthwise_relu (ReLU)  (None, 28, 28, 144)  0
['block_3_depthwise_BN[0][0]']

 block_3_project (Conv2D)       (None, 28, 28, 32)   4608
['block_3_depthwise_relu[0][0]']

 block_3_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_3_project[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_3_project_BN[0][0]']

 block_4_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_4_expand[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_4_expand_BN[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_4_expand_relu[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_4_depthwise[0][0]']
 malization)

 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_4_depthwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 28, 28, 32)   6144
['block_4_depthwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_4_project[0][0]']
 lization)

 block_4_add (Add)              (None, 28, 28, 32)   0
['block_3_project_BN[0][0]',
'block_4_project_BN[0][0]']

 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_4_add[0][0]']

 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_5_expand[0][0]']
 ization)
 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_5_expand_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_5_expand_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_5_depthwise[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_5_depthwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 28, 28, 32)   6144
['block_5_depthwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_5_project[0][0]']
 lization)

 block_5_add (Add)              (None, 28, 28, 32)   0
['block_4_add[0][0]',
'block_5_project_BN[0][0]']

 block_6_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_5_add[0][0]']

 block_6_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_6_expand[0][0]']
 ization)

 block_6_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_6_expand_BN[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 29, 29, 192)  0
['block_6_expand_relu[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 14, 14, 192)  1728
['block_6_pad[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 14, 14, 192)  768
['block_6_depthwise[0][0]']
 malization)

 block_6_depthwise_relu (ReLU)  (None, 14, 14, 192)  0
```

```
['block_6_depthwise_BN[0][0]']

 block_6_project (Conv2D)       (None, 14, 14, 64)   12288
['block_6_depthwise_relu[0][0]']

 block_6_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_6_project[0][0]']
 lization)

 block_7_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_6_project_BN[0][0]']

 block_7_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_7_expand[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_7_expand_BN[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_7_expand_relu[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_7_depthwise[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_7_depthwise_BN[0][0]']

 block_7_project (Conv2D)       (None, 14, 14, 64)   24576
['block_7_depthwise_relu[0][0]']

 block_7_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_7_project[0][0]']
 lization)

 block_7_add (Add)              (None, 14, 14, 64)   0
['block_6_project_BN[0][0]',
'block_7_project_BN[0][0]']

 block_8_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_7_add[0][0]']

 block_8_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_8_expand[0][0]']
 ization)
 block_8_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_8_expand_BN[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_8_expand_relu[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_8_depthwise[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_8_depthwise_BN[0][0]']

 block_8_project (Conv2D)       (None, 14, 14, 64)   24576
['block_8_depthwise_relu[0][0]']

 block_8_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_8_project[0][0]']
 lization)

 block_8_add (Add)              (None, 14, 14, 64)   0
['block_7_add[0][0]',
'block_8_project_BN[0][0]']

 block_9_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_8_add[0][0]']

 block_9_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_9_expand[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_9_expand_BN[0][0]']

 block_9_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_9_expand_relu[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_9_depthwise[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_9_depthwise_BN[0][0]']

 block_9_project (Conv2D)       (None, 14, 14, 64)   24576
['block_9_depthwise_relu[0][0]']
```

```
 block_9_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_9_project[0][0]']
 lization)

 block_9_add (Add)              (None, 14, 14, 64)   0
['block_8_add[0][0]',
'block_9_project_BN[0][0]']

 block_10_expand (Conv2D)       (None, 14, 14, 384)  24576
['block_9_add[0][0]']

 block_10_expand_BN (BatchNorma  (None, 14, 14, 384)  1536
['block_10_expand[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 14, 14, 384)  0
['block_10_expand_BN[0][0]']

 block_10_depthwise (DepthwiseC  (None, 14, 14, 384)  3456
['block_10_expand_relu[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 14, 14, 384)  1536
['block_10_depthwise[0][0]']
 rmalization)

 block_10_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_10_depthwise_BN[0][0]']

 block_10_project (Conv2D)      (None, 14, 14, 96)   36864
['block_10_depthwise_relu[0][0]']

 block_10_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_10_project[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_10_project_BN[0][0]']

 block_11_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_11_expand[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_11_expand_BN[0][0]']

 block_11_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_11_expand_relu[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_11_depthwise[0][0]']
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
['block_11_depthwise_BN[0][0]']

 block_11_project (Conv2D)      (None, 14, 14, 96)   55296
['block_11_depthwise_relu[0][0]']

 block_11_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_11_project[0][0]']
 alization)

 block_11_add (Add)             (None, 14, 14, 96)   0
['block_10_project_BN[0][0]',
'block_11_project_BN[0][0]']

 block_12_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_11_add[0][0]']

 block_12_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_12_expand[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_12_expand_BN[0][0]']

 block_12_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_12_expand_relu[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_12_depthwise[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
['block_12_depthwise_BN[0][0]']

 block_12_project (Conv2D)      (None, 14, 14, 96)   55296
['block_12_depthwise_relu[0][0]']

 block_12_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_12_project[0][0]']
 alization)
 block_12_add (Add)             (None, 14, 14, 96)   0
['block_11_add[0][0]',
'block_12_project_BN[0][0]']

 block_13_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_12_add[0][0]']

 block_13_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_13_expand[0][0]']
 lization)

 block_13_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_13_expand_BN[0][0]']

 block_13_pad (ZeroPadding2D)   (None, 15, 15, 576)  0
['block_13_expand_relu[0][0]']

 block_13_depthwise (DepthwiseC  (None, 7, 7, 576)   5184
['block_13_pad[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304
['block_13_depthwise[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 7, 7, 576)   0
['block_13_depthwise_BN[0][0]']

 block_13_project (Conv2D)      (None, 7, 7, 160)    92160
['block_13_depthwise_relu[0][0]']

 block_13_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_13_project[0][0]']
 alization)

 block_14_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_13_project_BN[0][0]']

 block_14_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_14_expand[0][0]']
 lization)

 block_14_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_14_expand_BN[0][0]']

 block_14_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_14_expand_relu[0][0]']
```

```
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_14_depthwise[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_14_depthwise_BN[0][0]']

 block_14_project (Conv2D)      (None, 7, 7, 160)    153600
['block_14_depthwise_relu[0][0]']

 block_14_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_14_project[0][0]']
 alization)

 block_14_add (Add)             (None, 7, 7, 160)    0
['block_13_project_BN[0][0]',
'block_14_project_BN[0][0]']

 block_15_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_14_add[0][0]']

 block_15_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_15_expand[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_15_expand_BN[0][0]']

 block_15_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_15_expand_relu[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_15_depthwise[0][0]']
 rmalization)

 block_15_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_15_depthwise_BN[0][0]']

 block_15_project (Conv2D)      (None, 7, 7, 160)    153600
['block_15_depthwise_relu[0][0]']

 block_15_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_15_project[0][0]']
 alization)
 block_15_add (Add)             (None, 7, 7, 160)    0
['block_14_add[0][0]',
'block_15_project_BN[0][0]']

 block_16_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_15_add[0][0]']

 block_16_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_16_expand[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_16_expand_BN[0][0]']

 block_16_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_16_expand_relu[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_16_depthwise[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_16_depthwise_BN[0][0]']

 block_16_project (Conv2D)      (None, 7, 7, 320)    307200
['block_16_depthwise_relu[0][0]']

 block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280
['block_16_project[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600
['block_16_project_BN[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120
['Conv_1[0][0]']

 out_relu (ReLU)                (None, 7, 7, 1280)   0
['Conv_1_bn[0][0]']

 global_average_pooling2d_2 (Gl  (None, 1280)        0
['out_relu[0][0]']
 obalAveragePooling2D)

 predictions (Dense)            (None, 12)           15372
['global_average_pooling2d_2[0][0
                                                                 ]']
```

```
================================================================================
==================
Total params: 2,273,356
Trainable params: 2,239,244
Non-trainable params: 34,112
--------------------------------------------------------------------------------
------------------
```

[ ]:

```python
# Call back 1:
base_learning_rate = 1e-4
opt1 = tf.keras.optimizers.Adam(learning_rate=base_learning_rate)
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='accuracy', min_delta=0, patience=20, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

# Call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1, callback_2]

#compiling our Model for dataset
mobilev2model.compile(optimizer=opt1,
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = mobilev2model.fit(
    train_mobilenet,
    epochs=60,
    validation_data=test_mobilenet,
    class_weight=class_weights,
    callbacks=callback_list)
```

```
Epoch 1/60
299/299 [==============================] - 42s 52ms/step - loss: 2.9365 -
accuracy: 0.4778 - val_loss: 4.0183 - val_accuracy: 0.1138 - lr: 1.0000e-04
Epoch 2/60
299/299 [==============================] - 15s 49ms/step - loss: 2.5710 -
accuracy: 0.5172 - val_loss: 3.9614 - val_accuracy: 0.0987 - lr: 1.0000e-04
Epoch 3/60
299/299 [==============================] - 15s 50ms/step - loss: 2.2262 -
```

```
accuracy: 0.5621 - val_loss: 3.2733 - val_accuracy: 0.1921 - lr: 1.0000e-04
Epoch 4/60
299/299 [==============================] - 15s 51ms/step - loss: 1.9482 -
accuracy: 0.6086 - val_loss: 2.4956 - val_accuracy: 0.3042 - lr: 1.0000e-04
Epoch 5/60
299/299 [==============================] - 15s 50ms/step - loss: 1.7147 -
accuracy: 0.6474 - val_loss: 2.7382 - val_accuracy: 0.2795 - lr: 1.0000e-04
Epoch 6/60
299/299 [==============================] - 15s 50ms/step - loss: 1.6062 -
accuracy: 0.6662 - val_loss: 2.5178 - val_accuracy: 0.3167 - lr: 1.0000e-04
Epoch 7/60
299/299 [==============================] - 15s 50ms/step - loss: 1.3325 -
accuracy: 0.7068 - val_loss: 3.4524 - val_accuracy: 0.2644 - lr: 1.0000e-04
Epoch 8/60
299/299 [==============================] - 15s 49ms/step - loss: 1.1919 -
accuracy: 0.7388 - val_loss: 3.5309 - val_accuracy: 0.3017 - lr: 1.0000e-04
Epoch 9/60
299/299 [==============================] - 15s 49ms/step - loss: 0.9422 -
accuracy: 0.7804 - val_loss: 2.1554 - val_accuracy: 0.4607 - lr: 1.0000e-05
Epoch 10/60
299/299 [==============================] - 15s 49ms/step - loss: 0.7704 -
accuracy: 0.8233 - val_loss: 2.0231 - val_accuracy: 0.4669 - lr: 1.0000e-05
Epoch 11/60
299/299 [==============================] - 15s 49ms/step - loss: 0.6823 -
accuracy: 0.8451 - val_loss: 1.9655 - val_accuracy: 0.4669 - lr: 1.0000e-05
Epoch 12/60
299/299 [==============================] - 15s 49ms/step - loss: 0.6214 -
accuracy: 0.8567 - val_loss: 1.9575 - val_accuracy: 0.4753 - lr: 1.0000e-05
Epoch 13/60
299/299 [==============================] - 15s 49ms/step - loss: 0.5892 -
accuracy: 0.8665 - val_loss: 1.9852 - val_accuracy: 0.4669 - lr: 1.0000e-05
Epoch 14/60
299/299 [==============================] - 15s 50ms/step - loss: 0.5322 -
accuracy: 0.8747 - val_loss: 1.9549 - val_accuracy: 0.4816 - lr: 1.0000e-05
Epoch 15/60
299/299 [==============================] - 15s 50ms/step - loss: 0.5058 -
accuracy: 0.8859 - val_loss: 1.9709 - val_accuracy: 0.4895 - lr: 1.0000e-05
Epoch 16/60
299/299 [==============================] - 15s 49ms/step - loss: 0.4584 -
accuracy: 0.8946 - val_loss: 1.9955 - val_accuracy: 0.4912 - lr: 1.0000e-05
Epoch 17/60
299/299 [==============================] - 15s 49ms/step - loss: 0.4394 -
accuracy: 0.8976 - val_loss: 1.9803 - val_accuracy: 0.4824 - lr: 1.0000e-05
Epoch 18/60
299/299 [==============================] - 15s 49ms/step - loss: 0.4141 -
accuracy: 0.9058 - val_loss: 2.0480 - val_accuracy: 0.4795 - lr: 1.0000e-05
Epoch 19/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3840 -
```

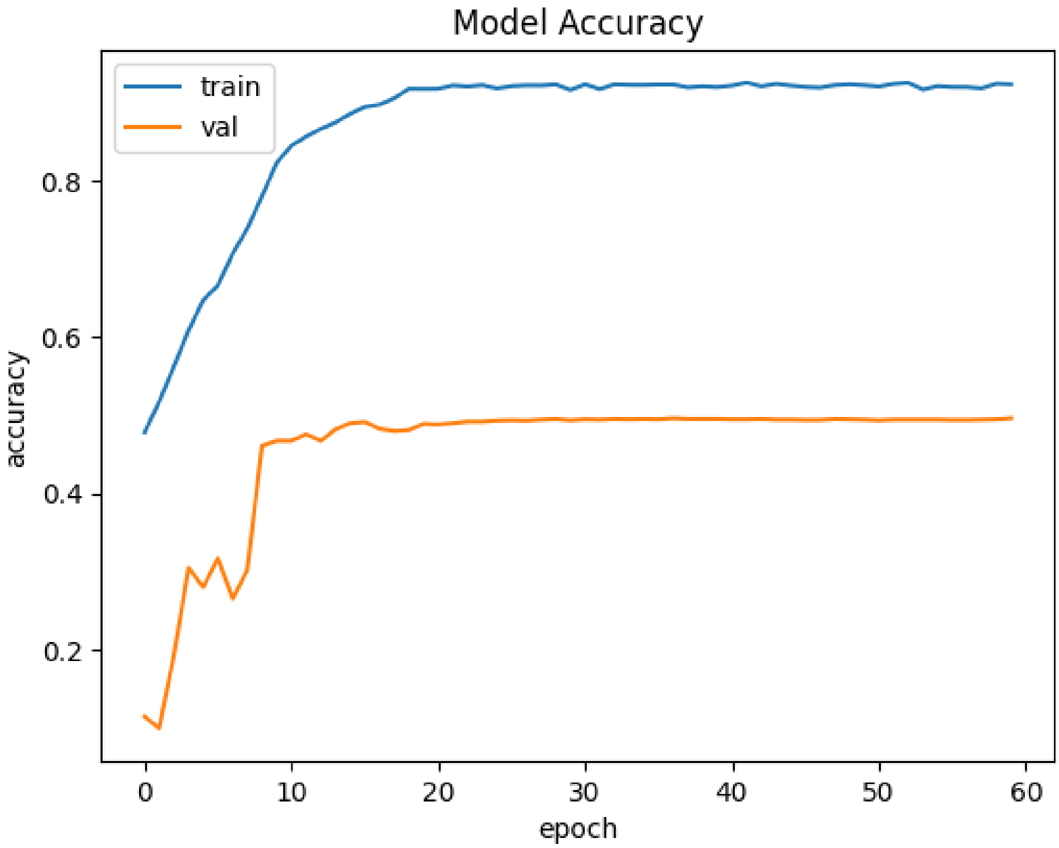
```
accuracy: 0.9180 - val_loss: 2.0139 - val_accuracy: 0.4808 - lr: 1.0000e-06
Epoch 20/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3761 -
accuracy: 0.9178 - val_loss: 2.0062 - val_accuracy: 0.4887 - lr: 1.0000e-06
Epoch 21/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3699 -
accuracy: 0.9180 - val_loss: 2.0107 - val_accuracy: 0.4879 - lr: 1.0000e-06
Epoch 22/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3650 -
accuracy: 0.9222 - val_loss: 2.0134 - val_accuracy: 0.4895 - lr: 1.0000e-06
Epoch 23/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3653 -
accuracy: 0.9208 - val_loss: 2.0148 - val_accuracy: 0.4916 - lr: 1.0000e-07
Epoch 24/60
299/299 [==============================] - 15s 50ms/step - loss: 0.3584 -
accuracy: 0.9226 - val_loss: 2.0156 - val_accuracy: 0.4916 - lr: 1.0000e-07
Epoch 25/60
299/299 [==============================] - 15s 50ms/step - loss: 0.3585 -
accuracy: 0.9184 - val_loss: 2.0167 - val_accuracy: 0.4929 - lr: 1.0000e-07
Epoch 26/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3668 -
accuracy: 0.9213 - val_loss: 2.0180 - val_accuracy: 0.4933 - lr: 1.0000e-07
Epoch 27/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3589 -
accuracy: 0.9223 - val_loss: 2.0190 - val_accuracy: 0.4929 - lr: 1.0000e-08
Epoch 28/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3616 -
accuracy: 0.9222 - val_loss: 2.0196 - val_accuracy: 0.4941 - lr: 1.0000e-08
Epoch 29/60
299/299 [==============================] - 15s 50ms/step - loss: 0.3633 -
accuracy: 0.9234 - val_loss: 2.0203 - val_accuracy: 0.4950 - lr: 1.0000e-08
Epoch 30/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3710 -
accuracy: 0.9165 - val_loss: 2.0202 - val_accuracy: 0.4933 - lr: 1.0000e-08
Epoch 31/60
299/299 [==============================] - 15s 50ms/step - loss: 0.3532 -
accuracy: 0.9235 - val_loss: 2.0203 - val_accuracy: 0.4946 - lr: 1.0000e-09
Epoch 32/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3706 -
accuracy: 0.9176 - val_loss: 2.0208 - val_accuracy: 0.4941 - lr: 1.0000e-09
Epoch 33/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3581 -
accuracy: 0.9235 - val_loss: 2.0212 - val_accuracy: 0.4950 - lr: 1.0000e-09
Epoch 34/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3680 -
accuracy: 0.9229 - val_loss: 2.0210 - val_accuracy: 0.4946 - lr: 1.0000e-09
Epoch 35/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3568 -
```

```
accuracy: 0.9228 - val_loss: 2.0207 - val_accuracy: 0.4950 - lr: 1.0000e-10
Epoch 36/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3540 -
accuracy: 0.9233 - val_loss: 2.0206 - val_accuracy: 0.4946 - lr: 1.0000e-10
Epoch 37/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3498 -
accuracy: 0.9233 - val_loss: 2.0207 - val_accuracy: 0.4958 - lr: 1.0000e-10
Epoch 38/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3690 -
accuracy: 0.9198 - val_loss: 2.0208 - val_accuracy: 0.4950 - lr: 1.0000e-10
Epoch 39/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3583 -
accuracy: 0.9211 - val_loss: 2.0211 - val_accuracy: 0.4950 - lr: 1.0000e-11
Epoch 40/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3699 -
accuracy: 0.9201 - val_loss: 2.0209 - val_accuracy: 0.4950 - lr: 1.0000e-11
Epoch 41/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3609 -
accuracy: 0.9220 - val_loss: 2.0213 - val_accuracy: 0.4946 - lr: 1.0000e-11
Epoch 42/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3533 -
accuracy: 0.9259 - val_loss: 2.0212 - val_accuracy: 0.4946 - lr: 1.0000e-11
Epoch 43/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3573 -
accuracy: 0.9210 - val_loss: 2.0214 - val_accuracy: 0.4950 - lr: 1.0000e-12
Epoch 44/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3560 -
accuracy: 0.9243 - val_loss: 2.0213 - val_accuracy: 0.4941 - lr: 1.0000e-12
Epoch 45/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3602 -
accuracy: 0.9221 - val_loss: 2.0207 - val_accuracy: 0.4941 - lr: 1.0000e-12
Epoch 46/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3602 -
accuracy: 0.9203 - val_loss: 2.0208 - val_accuracy: 0.4937 - lr: 1.0000e-12
Epoch 47/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3645 -
accuracy: 0.9196 - val_loss: 2.0211 - val_accuracy: 0.4937 - lr: 1.0000e-13
Epoch 48/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3614 -
accuracy: 0.9226 - val_loss: 2.0210 - val_accuracy: 0.4950 - lr: 1.0000e-13
Epoch 49/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3604 -
accuracy: 0.9236 - val_loss: 2.0214 - val_accuracy: 0.4946 - lr: 1.0000e-13
Epoch 50/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3581 -
accuracy: 0.9225 - val_loss: 2.0211 - val_accuracy: 0.4941 - lr: 1.0000e-13
Epoch 51/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3671 -
```

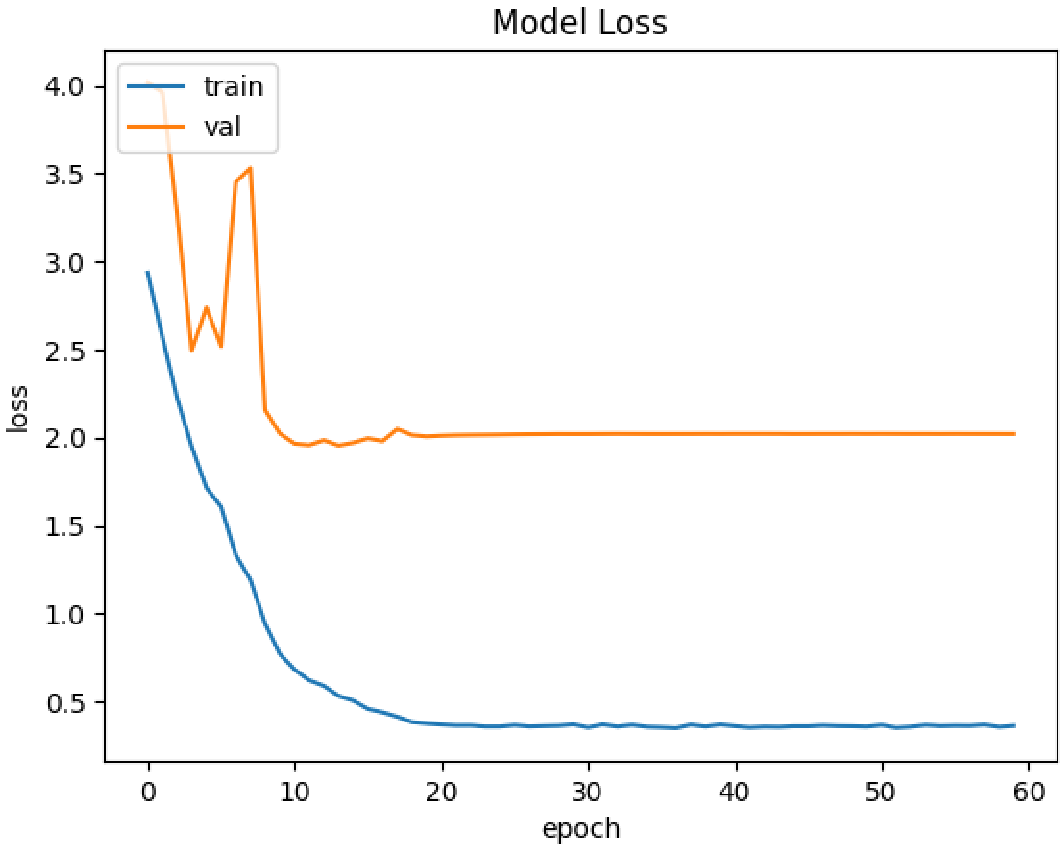
```
accuracy: 0.9208 - val_loss: 2.0214 - val_accuracy: 0.4933 - lr: 1.0000e-14
Epoch 52/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3518 -
accuracy: 0.9243 - val_loss: 2.0212 - val_accuracy: 0.4941 - lr: 1.0000e-14
Epoch 53/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3572 -
accuracy: 0.9257 - val_loss: 2.0209 - val_accuracy: 0.4941 - lr: 1.0000e-14
Epoch 54/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3670 -
accuracy: 0.9171 - val_loss: 2.0211 - val_accuracy: 0.4941 - lr: 1.0000e-14
Epoch 55/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3616 -
accuracy: 0.9213 - val_loss: 2.0208 - val_accuracy: 0.4941 - lr: 1.0000e-15
Epoch 56/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3634 -
accuracy: 0.9203 - val_loss: 2.0211 - val_accuracy: 0.4937 - lr: 1.0000e-15
Epoch 57/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3627 -
accuracy: 0.9202 - val_loss: 2.0210 - val_accuracy: 0.4937 - lr: 1.0000e-15
Epoch 58/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3689 -
accuracy: 0.9185 - val_loss: 2.0209 - val_accuracy: 0.4941 - lr: 1.0000e-15
Epoch 59/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3562 -
accuracy: 0.9245 - val_loss: 2.0208 - val_accuracy: 0.4946 - lr: 1.0000e-16
Epoch 60/60
299/299 [==============================] - 15s 49ms/step - loss: 0.3629 -
accuracy: 0.9236 - val_loss: 2.0206 - val_accuracy: 0.4958 - lr: 1.0000e-16
```

[ ]:

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



[ ]:
```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



```
[ ]: mobilev2model.save(os.path.
       ↪join(SAVE_DIR,'mobileNetV2_FilteredReducedClasses_noPretrain_noTuning.h5'))
```

Training MobileNetV3 With Reduced and Filtered Data Set

```
[ ]: mobilev3model = tf.keras.applications.MobileNetV3Large(
         input_shape=(224,224,3),
         alpha=1.0,
         include_top=True,
         weights=None,
         dropout_rate=0.8,
         pooling='max'
     )
     mobilev3model.summary()
```

```
Model: "MobilenetV3large"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_4 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 rescaling_1 (Rescaling)        (None, 224, 224, 3)  0
['input_4[0][0]']

 Conv (Conv2D)                  (None, 112, 112, 16  432
['rescaling_1[0][0]']
                                )

 Conv/BatchNorm (BatchNormaliza  (None, 112, 112, 16  64          ['Conv[0][0]']
 tion)                          )

 tf.__operators__.add_29 (TFOpL  (None, 112, 112, 16  0
['Conv/BatchNorm[0][0]']
 ambda)                         )

 re_lu_40 (ReLU)                (None, 112, 112, 16  0
['tf.__operators__.add_29[0][0]']
                                )

 tf.math.multiply_29 (TFOpLambd  (None, 112, 112, 16  0
['re_lu_40[0][0]']
 a)                             )

 multiply_21 (Multiply)         (None, 112, 112, 16  0
['Conv/BatchNorm[0][0]',
                                )
'tf.math.multiply_29[0][0]']

 expanded_conv/depthwise (Depth  (None, 112, 112, 16  144
['multiply_21[0][0]']
 wiseConv2D)                    )

 expanded_conv/depthwise/BatchN  (None, 112, 112, 16  64
['expanded_conv/depthwise[0][0]']
 orm (BatchNormalization)       )

 re_lu_41 (ReLU)                (None, 112, 112, 16  0
['expanded_conv/depthwise/BatchNo
                                )                               rm[0][0]']

 expanded_conv/project (Conv2D)  (None, 112, 112, 16  256
['re_lu_41[0][0]']
                                )

 expanded_conv/project/BatchNor  (None, 112, 112, 16  64
['expanded_conv/project[0][0]']
 m (BatchNormalization)         )

 expanded_conv/Add (Add)        (None, 112, 112, 16  0
['multiply_21[0][0]',
                                )
'expanded_conv/project/BatchNorm
                                                                 [0][0]']

 expanded_conv_1/expand (Conv2D  (None, 112, 112, 64  1024
['expanded_conv/Add[0][0]']
 )                              )

 expanded_conv_1/expand/BatchNo  (None, 112, 112, 64  256
['expanded_conv_1/expand[0][0]']
 rm (BatchNormalization)        )

 re_lu_42 (ReLU)                (None, 112, 112, 64  0
['expanded_conv_1/expand/BatchNor
                                )                                m[0][0]']

 expanded_conv_1/depthwise/pad   (None, 113, 113, 64  0
['re_lu_42[0][0]']
 (ZeroPadding2D)                )

 expanded_conv_1/depthwise (Dep  (None, 56, 56, 64)  576
['expanded_conv_1/depthwise/pad[0
 thwiseConv2D)                                                   ][0]']

 expanded_conv_1/depthwise/Batc  (None, 56, 56, 64)  256
['expanded_conv_1/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_43 (ReLU)                (None, 56, 56, 64)   0
['expanded_conv_1/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_1/project (Conv2  (None, 56, 56, 24)  1536
['re_lu_43[0][0]']
 D)

 expanded_conv_1/project/BatchN  (None, 56, 56, 24)  96
['expanded_conv_1/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_2/expand (Conv2D  (None, 56, 56, 72)  1728
['expanded_conv_1/project/BatchNo
 )                                                               rm[0][0]']
 expanded_conv_2/expand/BatchNo  (None, 56, 56, 72)  288
['expanded_conv_2/expand[0][0]']
 rm (BatchNormalization)

 re_lu_44 (ReLU)                (None, 56, 56, 72)   0
['expanded_conv_2/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_2/depthwise (Dep  (None, 56, 56, 72)  648
['re_lu_44[0][0]']
 thwiseConv2D)

 expanded_conv_2/depthwise/Batc  (None, 56, 56, 72)  288
['expanded_conv_2/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_45 (ReLU)                (None, 56, 56, 72)   0
['expanded_conv_2/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_2/project (Conv2  (None, 56, 56, 24)  1728
['re_lu_45[0][0]']
 D)

 expanded_conv_2/project/BatchN  (None, 56, 56, 24)  96
['expanded_conv_2/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_2/Add (Add)      (None, 56, 56, 24)   0
['expanded_conv_1/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_2/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_3/expand (Conv2D  (None, 56, 56, 72)  1728
['expanded_conv_2/Add[0][0]']
 )

 expanded_conv_3/expand/BatchNo  (None, 56, 56, 72)  288
['expanded_conv_3/expand[0][0]']
 rm (BatchNormalization)

 re_lu_46 (ReLU)                (None, 56, 56, 72)   0
['expanded_conv_3/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_3/depthwise/pad   (None, 59, 59, 72)  0
```

```
['re_lu_46[0][0]']
 (ZeroPadding2D)

 expanded_conv_3/depthwise (Dep  (None, 28, 28, 72)  1800
['expanded_conv_3/depthwise/pad[0
 thwiseConv2D)                                                 ][0]']

 expanded_conv_3/depthwise/Batc  (None, 28, 28, 72)  288
['expanded_conv_3/depthwise[0][0]
 hNorm (BatchNormalization)                                    ']

 re_lu_47 (ReLU)                (None, 28, 28, 72)   0
['expanded_conv_3/depthwise/Batch
                                                               Norm[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 72)    0
['re_lu_47[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_3/squeeze_excite  (None, 1, 1, 24)    1752
['expanded_conv_3/squeeze_excite/
 /Conv (Conv2D)                                                AvgPool[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 24)    0
['expanded_conv_3/squeeze_excite/
 /Relu (ReLU)                                                  Conv[0][0]']

 expanded_conv_3/squeeze_excite  (None, 1, 1, 72)    1800
['expanded_conv_3/squeeze_excite/
 /Conv_1 (Conv2D)                                              Relu[0][0]']

 tf.__operators__.add_30 (TFOpL  (None, 1, 1, 72)    0
['expanded_conv_3/squeeze_excite/
 ambda)                                                        Conv_1[0][0]']

 re_lu_48 (ReLU)                (None, 1, 1, 72)     0
['tf.__operators__.add_30[0][0]']

 tf.math.multiply_30 (TFOpLambd  (None, 1, 1, 72)    0
['re_lu_48[0][0]']
 a)

 expanded_conv_3/squeeze_excite  (None, 28, 28, 72)  0
['re_lu_47[0][0]',
 /Mul (Multiply)
'tf.math.multiply_30[0][0]']
```

```
 expanded_conv_3/project (Conv2  (None, 28, 28, 40)  2880
['expanded_conv_3/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_3/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_3/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_4/expand (Conv2D  (None, 28, 28, 120)  4800
['expanded_conv_3/project/BatchNo
 )                                                               rm[0][0]']

 expanded_conv_4/expand/BatchNo  (None, 28, 28, 120)  480
['expanded_conv_4/expand[0][0]']
 rm (BatchNormalization)

 re_lu_49 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_4/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_4/depthwise (Dep  (None, 28, 28, 120)  3000
['re_lu_49[0][0]']
 thwiseConv2D)

 expanded_conv_4/depthwise/Batc  (None, 28, 28, 120)  480
['expanded_conv_4/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_50 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_4/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 120)   0
['re_lu_50[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_4/squeeze_excite  (None, 1, 1, 32)    3872
['expanded_conv_4/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 32)    0
['expanded_conv_4/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_4/squeeze_excite  (None, 1, 1, 120)   3960
['expanded_conv_4/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']
 tf.__operators__.add_31 (TFOpL  (None, 1, 1, 120)   0
['expanded_conv_4/squeeze_excite/
 ambda)                                                          Conv_1[0][0]']

 re_lu_51 (ReLU)                (None, 1, 1, 120)    0
['tf.__operators__.add_31[0][0]']

 tf.math.multiply_31 (TFOpLambd  (None, 1, 1, 120)   0
['re_lu_51[0][0]']
 a)

 expanded_conv_4/squeeze_excite  (None, 28, 28, 120)  0
['re_lu_50[0][0]',
 /Mul (Multiply)
'tf.math.multiply_31[0][0]']

 expanded_conv_4/project (Conv2  (None, 28, 28, 40)  4800
['expanded_conv_4/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_4/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_4/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_4/Add (Add)      (None, 28, 28, 40)   0
['expanded_conv_3/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_4/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_5/expand (Conv2D  (None, 28, 28, 120)  4800
['expanded_conv_4/Add[0][0]']
 )

 expanded_conv_5/expand/BatchNo  (None, 28, 28, 120)  480
['expanded_conv_5/expand[0][0]']
 rm (BatchNormalization)

 re_lu_52 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_5/expand/BatchNor
                                                                 m[0][0]']

 expanded_conv_5/depthwise (Dep  (None, 28, 28, 120)  3000
['re_lu_52[0][0]']
 thwiseConv2D)

 expanded_conv_5/depthwise/Batc  (None, 28, 28, 120)  480
```

```
['expanded_conv_5/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 re_lu_53 (ReLU)                (None, 28, 28, 120)  0
['expanded_conv_5/depthwise/Batch
                                                                 Norm[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 120)   0
['re_lu_53[0][0]']
 /AvgPool (GlobalAveragePooling
 2D)

 expanded_conv_5/squeeze_excite  (None, 1, 1, 32)    3872
['expanded_conv_5/squeeze_excite/
 /Conv (Conv2D)                                                  AvgPool[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 32)    0
['expanded_conv_5/squeeze_excite/
 /Relu (ReLU)                                                    Conv[0][0]']

 expanded_conv_5/squeeze_excite  (None, 1, 1, 120)   3960
['expanded_conv_5/squeeze_excite/
 /Conv_1 (Conv2D)                                                Relu[0][0]']

 tf.__operators__.add_32 (TFOpL  (None, 1, 1, 120)   0
['expanded_conv_5/squeeze_excite/
 ambda)                                                          Conv_1[0][0]']

 re_lu_54 (ReLU)                (None, 1, 1, 120)    0
['tf.__operators__.add_32[0][0]']

 tf.math.multiply_32 (TFOpLambd  (None, 1, 1, 120)   0
['re_lu_54[0][0]']
 a)

 expanded_conv_5/squeeze_excite  (None, 28, 28, 120)  0
['re_lu_53[0][0]',
 /Mul (Multiply)
'tf.math.multiply_32[0][0]']

 expanded_conv_5/project (Conv2  (None, 28, 28, 40)  4800
['expanded_conv_5/squeeze_excite/
 D)                                                              Mul[0][0]']

 expanded_conv_5/project/BatchN  (None, 28, 28, 40)  160
['expanded_conv_5/project[0][0]']
 orm (BatchNormalization)
 expanded_conv_5/Add (Add)      (None, 28, 28, 40)   0
['expanded_conv_4/Add[0][0]',
'expanded_conv_5/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_6/expand (Conv2D  (None, 28, 28, 240)  9600
['expanded_conv_5/Add[0][0]']
 )

 expanded_conv_6/expand/BatchNo  (None, 28, 28, 240)  960
['expanded_conv_6/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_33 (TFOpL  (None, 28, 28, 240)  0
['expanded_conv_6/expand/BatchNor
 ambda)                                                          m[0][0]']

 re_lu_55 (ReLU)                (None, 28, 28, 240)  0
['tf.__operators__.add_33[0][0]']

 tf.math.multiply_33 (TFOpLambd  (None, 28, 28, 240)  0
['re_lu_55[0][0]']
 a)

 multiply_22 (Multiply)         (None, 28, 28, 240)  0
['expanded_conv_6/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_33[0][0]']

 expanded_conv_6/depthwise/pad   (None, 29, 29, 240)  0
['multiply_22[0][0]']
 (ZeroPadding2D)

 expanded_conv_6/depthwise (Dep  (None, 14, 14, 240)  2160
['expanded_conv_6/depthwise/pad[0
 thwiseConv2D)                                                   ][0]']

 expanded_conv_6/depthwise/Batc  (None, 14, 14, 240)  960
['expanded_conv_6/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 tf.__operators__.add_34 (TFOpL  (None, 14, 14, 240)  0
['expanded_conv_6/depthwise/Batch
 ambda)                                                          Norm[0][0]']

 re_lu_56 (ReLU)                (None, 14, 14, 240)  0
['tf.__operators__.add_34[0][0]']
```

```
 tf.math.multiply_34 (TFOpLambd  (None, 14, 14, 240)  0
['re_lu_56[0][0]']
 a)

 multiply_23 (Multiply)         (None, 14, 14, 240)  0
['expanded_conv_6/depthwise/Batch
                                                                Norm[0][0]',
'tf.math.multiply_34[0][0]']

 expanded_conv_6/project (Conv2  (None, 14, 14, 80)  19200
['multiply_23[0][0]']
 D)

 expanded_conv_6/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_6/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_7/expand (Conv2D  (None, 14, 14, 200)  16000
['expanded_conv_6/project/BatchNo
 )                                                              rm[0][0]']

 expanded_conv_7/expand/BatchNo  (None, 14, 14, 200)  800
['expanded_conv_7/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_35 (TFOpL  (None, 14, 14, 200)  0
['expanded_conv_7/expand/BatchNor
 ambda)                                                         m[0][0]']

 re_lu_57 (ReLU)                (None, 14, 14, 200)  0
['tf.__operators__.add_35[0][0]']

 tf.math.multiply_35 (TFOpLambd  (None, 14, 14, 200)  0
['re_lu_57[0][0]']
 a)

 multiply_24 (Multiply)         (None, 14, 14, 200)  0
['expanded_conv_7/expand/BatchNor
                                                                m[0][0]',
'tf.math.multiply_35[0][0]']

 expanded_conv_7/depthwise (Dep  (None, 14, 14, 200)  1800
['multiply_24[0][0]']
 thwiseConv2D)

 expanded_conv_7/depthwise/Batc  (None, 14, 14, 200)  800
['expanded_conv_7/depthwise[0][0]
 hNorm (BatchNormalization)                                     ']
 tf.__operators__.add_36 (TFOpL  (None, 14, 14, 200)  0
['expanded_conv_7/depthwise/Batch
 ambda)                                                          Norm[0][0]']

 re_lu_58 (ReLU)                (None, 14, 14, 200)  0
['tf.__operators__.add_36[0][0]']

 tf.math.multiply_36 (TFOpLambd  (None, 14, 14, 200)  0
['re_lu_58[0][0]']
 a)

 multiply_25 (Multiply)         (None, 14, 14, 200)  0
['expanded_conv_7/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_36[0][0]']

 expanded_conv_7/project (Conv2  (None, 14, 14, 80)  16000
['multiply_25[0][0]']
 D)

 expanded_conv_7/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_7/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_7/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_6/project/BatchNo
                                                                 rm[0][0]',
'expanded_conv_7/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_8/expand (Conv2D  (None, 14, 14, 184)  14720
['expanded_conv_7/Add[0][0]']
 )

 expanded_conv_8/expand/BatchNo  (None, 14, 14, 184)  736
['expanded_conv_8/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_37 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_8/expand/BatchNor
 ambda)                                                          m[0][0]']

 re_lu_59 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_37[0][0]']

 tf.math.multiply_37 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_59[0][0]']
```

```
 a)

 multiply_26 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_8/expand/BatchNor
                                                                 m[0][0]',
'tf.math.multiply_37[0][0]']

 expanded_conv_8/depthwise (Dep  (None, 14, 14, 184)  1656
['multiply_26[0][0]']
 thwiseConv2D)

 expanded_conv_8/depthwise/Batc  (None, 14, 14, 184)  736
['expanded_conv_8/depthwise[0][0]
 hNorm (BatchNormalization)                                      ']

 tf.__operators__.add_38 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_8/depthwise/Batch
 ambda)                                                          Norm[0][0]']

 re_lu_60 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_38[0][0]']

 tf.math.multiply_38 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_60[0][0]']
 a)

 multiply_27 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_8/depthwise/Batch
                                                                 Norm[0][0]',
'tf.math.multiply_38[0][0]']

 expanded_conv_8/project (Conv2  (None, 14, 14, 80)  14720
['multiply_27[0][0]']
 D)

 expanded_conv_8/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_8/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_8/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_7/Add[0][0]',
'expanded_conv_8/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_9/expand (Conv2D  (None, 14, 14, 184)  14720
['expanded_conv_8/Add[0][0]']
 )
 expanded_conv_9/expand/BatchNo  (None, 14, 14, 184)  736
['expanded_conv_9/expand[0][0]']
 rm (BatchNormalization)

 tf.__operators__.add_39 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_9/expand/BatchNor
 ambda)                                                           m[0][0]']

 re_lu_61 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_39[0][0]']

 tf.math.multiply_39 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_61[0][0]']
 a)

 multiply_28 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_9/expand/BatchNor
                                                                  m[0][0]',
'tf.math.multiply_39[0][0]']

 expanded_conv_9/depthwise (Dep  (None, 14, 14, 184)  1656
['multiply_28[0][0]']
 thwiseConv2D)

 expanded_conv_9/depthwise/Batc  (None, 14, 14, 184)  736
['expanded_conv_9/depthwise[0][0]
 hNorm (BatchNormalization)                                       ']

 tf.__operators__.add_40 (TFOpL  (None, 14, 14, 184)  0
['expanded_conv_9/depthwise/Batch
 ambda)                                                           Norm[0][0]']

 re_lu_62 (ReLU)                (None, 14, 14, 184)  0
['tf.__operators__.add_40[0][0]']

 tf.math.multiply_40 (TFOpLambd  (None, 14, 14, 184)  0
['re_lu_62[0][0]']
 a)

 multiply_29 (Multiply)         (None, 14, 14, 184)  0
['expanded_conv_9/depthwise/Batch
                                                                  Norm[0][0]',
'tf.math.multiply_40[0][0]']

 expanded_conv_9/project (Conv2  (None, 14, 14, 80)  14720
['multiply_29[0][0]']
 D)
 expanded_conv_9/project/BatchN  (None, 14, 14, 80)  320
['expanded_conv_9/project[0][0]']
 orm (BatchNormalization)

 expanded_conv_9/Add (Add)      (None, 14, 14, 80)   0
['expanded_conv_8/Add[0][0]',
'expanded_conv_9/project/BatchNo
                                                                 rm[0][0]']

 expanded_conv_10/expand (Conv2  (None, 14, 14, 480)  38400
['expanded_conv_9/Add[0][0]']
 D)

 expanded_conv_10/expand/BatchN  (None, 14, 14, 480)  1920
['expanded_conv_10/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_41 (TFOpL  (None, 14, 14, 480)  0
['expanded_conv_10/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_63 (ReLU)                (None, 14, 14, 480)  0
['tf.__operators__.add_41[0][0]']

 tf.math.multiply_41 (TFOpLambd  (None, 14, 14, 480)  0
['re_lu_63[0][0]']
 a)

 multiply_30 (Multiply)         (None, 14, 14, 480)  0
['expanded_conv_10/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_41[0][0]']

 expanded_conv_10/depthwise (De  (None, 14, 14, 480)  4320
['multiply_30[0][0]']
 pthwiseConv2D)

 expanded_conv_10/depthwise/Bat  (None, 14, 14, 480)  1920
['expanded_conv_10/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_42 (TFOpL  (None, 14, 14, 480)  0
['expanded_conv_10/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_64 (ReLU)                (None, 14, 14, 480)  0
['tf.__operators__.add_42[0][0]']
```

```
 tf.math.multiply_42 (TFOpLambd  (None, 14, 14, 480)  0
['re_lu_64[0][0]']
 a)

 multiply_31 (Multiply)         (None, 14, 14, 480)  0
['expanded_conv_10/depthwise/Batc
                                                                hNorm[0][0]',
'tf.math.multiply_42[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 480)   0
['multiply_31[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_10/squeeze_excit  (None, 1, 1, 120)   57720
['expanded_conv_10/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 120)   0
['expanded_conv_10/squeeze_excite
 e/Relu (ReLU)                                                  /Conv[0][0]']

 expanded_conv_10/squeeze_excit  (None, 1, 1, 480)   58080
['expanded_conv_10/squeeze_excite
 e/Conv_1 (Conv2D)                                              /Relu[0][0]']

 tf.__operators__.add_43 (TFOpL  (None, 1, 1, 480)   0
['expanded_conv_10/squeeze_excite
 ambda)                                                         /Conv_1[0][0]']

 re_lu_65 (ReLU)                (None, 1, 1, 480)    0
['tf.__operators__.add_43[0][0]']

 tf.math.multiply_43 (TFOpLambd  (None, 1, 1, 480)   0
['re_lu_65[0][0]']
 a)

 expanded_conv_10/squeeze_excit  (None, 14, 14, 480)  0
['multiply_31[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_43[0][0]']

 expanded_conv_10/project (Conv  (None, 14, 14, 112)  53760
['expanded_conv_10/squeeze_excite
 2D)                                                            /Mul[0][0]']

 expanded_conv_10/project/Batch  (None, 14, 14, 112)  448
```

```
['expanded_conv_10/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_11/expand (Conv2  (None, 14, 14, 672)  75264
['expanded_conv_10/project/BatchN
 D)                                                              orm[0][0]']

 expanded_conv_11/expand/BatchN  (None, 14, 14, 672)  2688
['expanded_conv_11/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_44 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_11/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_66 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_44[0][0]']

 tf.math.multiply_44 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_66[0][0]']
 a)

 multiply_32 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_11/expand/BatchNo
                                                                 rm[0][0]',

'tf.math.multiply_44[0][0]']

 expanded_conv_11/depthwise (De  (None, 14, 14, 672)  6048
['multiply_32[0][0]']
 pthwiseConv2D)

 expanded_conv_11/depthwise/Bat  (None, 14, 14, 672)  2688
['expanded_conv_11/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_45 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_11/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_67 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_45[0][0]']

 tf.math.multiply_45 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_67[0][0]']
 a)

 multiply_33 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_11/depthwise/Batc
```

```
                                                                 hNorm[0][0]',
'tf.math.multiply_45[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 672)   0
['multiply_33[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_11/squeeze_excit  (None, 1, 1, 168)   113064
['expanded_conv_11/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 168)   0
['expanded_conv_11/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_11/squeeze_excit  (None, 1, 1, 672)   113568
['expanded_conv_11/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_46 (TFOpL  (None, 1, 1, 672)   0
['expanded_conv_11/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_68 (ReLU)                (None, 1, 1, 672)    0
['tf.__operators__.add_46[0][0]']

 tf.math.multiply_46 (TFOpLambd  (None, 1, 1, 672)   0
['re_lu_68[0][0]']
 a)

 expanded_conv_11/squeeze_excit  (None, 14, 14, 672)  0
['multiply_33[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_46[0][0]']

 expanded_conv_11/project (Conv  (None, 14, 14, 112)  75264
['expanded_conv_11/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_11/project/Batch  (None, 14, 14, 112)  448
['expanded_conv_11/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_11/Add (Add)     (None, 14, 14, 112)  0
['expanded_conv_10/project/BatchN
                                                                 orm[0][0]',
'expanded_conv_11/project/BatchN
                                                                 orm[0][0]']

 expanded_conv_12/expand (Conv2  (None, 14, 14, 672)  75264
['expanded_conv_11/Add[0][0]']
 D)

 expanded_conv_12/expand/BatchN  (None, 14, 14, 672)  2688
['expanded_conv_12/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_47 (TFOpL  (None, 14, 14, 672)  0
['expanded_conv_12/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_69 (ReLU)                (None, 14, 14, 672)  0
['tf.__operators__.add_47[0][0]']

 tf.math.multiply_47 (TFOpLambd  (None, 14, 14, 672)  0
['re_lu_69[0][0]']
 a)

 multiply_34 (Multiply)         (None, 14, 14, 672)  0
['expanded_conv_12/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_47[0][0]']

 expanded_conv_12/depthwise/pad  (None, 17, 17, 672)  0
['multiply_34[0][0]']
  (ZeroPadding2D)

 expanded_conv_12/depthwise (De  (None, 7, 7, 672)   16800
['expanded_conv_12/depthwise/pad[
 pthwiseConv2D)                                                  0][0]']

 expanded_conv_12/depthwise/Bat  (None, 7, 7, 672)   2688
['expanded_conv_12/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_48 (TFOpL  (None, 7, 7, 672)   0
['expanded_conv_12/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_70 (ReLU)                (None, 7, 7, 672)    0
['tf.__operators__.add_48[0][0]']

 tf.math.multiply_48 (TFOpLambd  (None, 7, 7, 672)   0
['re_lu_70[0][0]']
```

```
 a)

 multiply_35 (Multiply)         (None, 7, 7, 672)    0
['expanded_conv_12/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_48[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 672)   0
['multiply_35[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_12/squeeze_excit  (None, 1, 1, 168)   113064
['expanded_conv_12/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 168)   0
['expanded_conv_12/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_12/squeeze_excit  (None, 1, 1, 672)   113568
['expanded_conv_12/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_49 (TFOpL  (None, 1, 1, 672)   0
['expanded_conv_12/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_71 (ReLU)                (None, 1, 1, 672)    0
['tf.__operators__.add_49[0][0]']

 tf.math.multiply_49 (TFOpLambd  (None, 1, 1, 672)   0
['re_lu_71[0][0]']
 a)

 expanded_conv_12/squeeze_excit  (None, 7, 7, 672)   0
['multiply_35[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_49[0][0]']

 expanded_conv_12/project (Conv  (None, 7, 7, 160)   107520
['expanded_conv_12/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_12/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_12/project[0][0]'
 Norm (BatchNormalization)                                       ]
 expanded_conv_13/expand (Conv2  (None, 7, 7, 960)   153600
['expanded_conv_12/project/BatchN
 D)                                                              orm[0][0]']

 expanded_conv_13/expand/BatchN  (None, 7, 7, 960)   3840
['expanded_conv_13/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_50 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_13/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_72 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_50[0][0]']

 tf.math.multiply_50 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_72[0][0]']
 a)

 multiply_36 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_13/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_50[0][0]']

 expanded_conv_13/depthwise (De  (None, 7, 7, 960)   24000
['multiply_36[0][0]']
 pthwiseConv2D)

 expanded_conv_13/depthwise/Bat  (None, 7, 7, 960)   3840
['expanded_conv_13/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_51 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_13/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_73 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_51[0][0]']

 tf.math.multiply_51 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_73[0][0]']
 a)

 multiply_37 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_13/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_51[0][0]']
```

```
 expanded_conv_13/squeeze_excit  (None, 1, 1, 960)   0
['multiply_37[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_13/squeeze_excit  (None, 1, 1, 240)   230640
['expanded_conv_13/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_13/squeeze_excit  (None, 1, 1, 240)   0
['expanded_conv_13/squeeze_excite
 e/Relu (ReLU)                                                    /Conv[0][0]']

 expanded_conv_13/squeeze_excit  (None, 1, 1, 960)   231360
['expanded_conv_13/squeeze_excite
 e/Conv_1 (Conv2D)                                                /Relu[0][0]']

 tf.__operators__.add_52 (TFOpL  (None, 1, 1, 960)   0
['expanded_conv_13/squeeze_excite
 ambda)                                                           /Conv_1[0][0]']

 re_lu_74 (ReLU)                (None, 1, 1, 960)    0
['tf.__operators__.add_52[0][0]']

 tf.math.multiply_52 (TFOpLambd  (None, 1, 1, 960)   0
['re_lu_74[0][0]']
 a)

 expanded_conv_13/squeeze_excit  (None, 7, 7, 960)   0
['multiply_37[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_52[0][0]']

 expanded_conv_13/project (Conv  (None, 7, 7, 160)   153600
['expanded_conv_13/squeeze_excite
 2D)                                                              /Mul[0][0]']

 expanded_conv_13/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_13/project[0][0]'
 Norm (BatchNormalization)                                        ]

 expanded_conv_13/Add (Add)     (None, 7, 7, 160)    0
['expanded_conv_12/project/BatchN
                                                                  orm[0][0]',
'expanded_conv_13/project/BatchN
                                                                  orm[0][0]']
 expanded_conv_14/expand (Conv2  (None, 7, 7, 960)   153600
['expanded_conv_13/Add[0][0]']
 D)

 expanded_conv_14/expand/BatchN  (None, 7, 7, 960)   3840
['expanded_conv_14/expand[0][0]']
 orm (BatchNormalization)

 tf.__operators__.add_53 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_14/expand/BatchNo
 ambda)                                                          rm[0][0]']

 re_lu_75 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_53[0][0]']

 tf.math.multiply_53 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_75[0][0]']
 a)

 multiply_38 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_14/expand/BatchNo
                                                                 rm[0][0]',
'tf.math.multiply_53[0][0]']

 expanded_conv_14/depthwise (De  (None, 7, 7, 960)   24000
['multiply_38[0][0]']
 pthwiseConv2D)

 expanded_conv_14/depthwise/Bat  (None, 7, 7, 960)   3840
['expanded_conv_14/depthwise[0][0
 chNorm (BatchNormalization)                                     ]']

 tf.__operators__.add_54 (TFOpL  (None, 7, 7, 960)   0
['expanded_conv_14/depthwise/Batc
 ambda)                                                          hNorm[0][0]']

 re_lu_76 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_54[0][0]']

 tf.math.multiply_54 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_76[0][0]']
 a)

 multiply_39 (Multiply)         (None, 7, 7, 960)    0
['expanded_conv_14/depthwise/Batc
                                                                 hNorm[0][0]',
'tf.math.multiply_54[0][0]']
```

```
 expanded_conv_14/squeeze_excit  (None, 1, 1, 960)   0
['multiply_39[0][0]']
 e/AvgPool (GlobalAveragePoolin
 g2D)

 expanded_conv_14/squeeze_excit  (None, 1, 1, 240)   230640
['expanded_conv_14/squeeze_excite
 e/Conv (Conv2D)
/AvgPool[0][0]']

 expanded_conv_14/squeeze_excit  (None, 1, 1, 240)   0
['expanded_conv_14/squeeze_excite
 e/Relu (ReLU)                                                   /Conv[0][0]']

 expanded_conv_14/squeeze_excit  (None, 1, 1, 960)   231360
['expanded_conv_14/squeeze_excite
 e/Conv_1 (Conv2D)                                               /Relu[0][0]']

 tf.__operators__.add_55 (TFOpL  (None, 1, 1, 960)   0
['expanded_conv_14/squeeze_excite
 ambda)                                                          /Conv_1[0][0]']

 re_lu_77 (ReLU)                (None, 1, 1, 960)    0
['tf.__operators__.add_55[0][0]']

 tf.math.multiply_55 (TFOpLambd  (None, 1, 1, 960)   0
['re_lu_77[0][0]']
 a)

 expanded_conv_14/squeeze_excit  (None, 7, 7, 960)   0
['multiply_39[0][0]',
 e/Mul (Multiply)
'tf.math.multiply_55[0][0]']

 expanded_conv_14/project (Conv  (None, 7, 7, 160)   153600
['expanded_conv_14/squeeze_excite
 2D)                                                             /Mul[0][0]']

 expanded_conv_14/project/Batch  (None, 7, 7, 160)   640
['expanded_conv_14/project[0][0]'
 Norm (BatchNormalization)                                       ]

 expanded_conv_14/Add (Add)     (None, 7, 7, 160)    0
['expanded_conv_13/Add[0][0]',
'expanded_conv_14/project/BatchN
                                                                 orm[0][0]']
 Conv_1 (Conv2D)                (None, 7, 7, 960)    153600
['expanded_conv_14/Add[0][0]']

 Conv_1/BatchNorm (BatchNormali  (None, 7, 7, 960)   3840
['Conv_1[0][0]']
 zation)

 tf.__operators__.add_56 (TFOpL  (None, 7, 7, 960)   0
['Conv_1/BatchNorm[0][0]']
 ambda)

 re_lu_78 (ReLU)                (None, 7, 7, 960)    0
['tf.__operators__.add_56[0][0]']

 tf.math.multiply_56 (TFOpLambd  (None, 7, 7, 960)   0
['re_lu_78[0][0]']
 a)

 multiply_40 (Multiply)         (None, 7, 7, 960)    0
['Conv_1/BatchNorm[0][0]',
'tf.math.multiply_56[0][0]']

 global_average_pooling2d_3 (Gl  (None, 1, 1, 960)   0
['multiply_40[0][0]']
 obalAveragePooling2D)

 Conv_2 (Conv2D)                (None, 1, 1, 1280)   1230080
['global_average_pooling2d_3[0][0
                                                                 ]']

 tf.__operators__.add_57 (TFOpL  (None, 1, 1, 1280)  0
['Conv_2[0][0]']
 ambda)

 re_lu_79 (ReLU)                (None, 1, 1, 1280)   0
['tf.__operators__.add_57[0][0]']

 tf.math.multiply_57 (TFOpLambd  (None, 1, 1, 1280)  0
['re_lu_79[0][0]']
 a)

 multiply_41 (Multiply)         (None, 1, 1, 1280)   0
['Conv_2[0][0]',
'tf.math.multiply_57[0][0]']

 dropout_1 (Dropout)            (None, 1, 1, 1280)   0
['multiply_41[0][0]']
```

```
 Logits (Conv2D)                (None, 1, 1, 1000)   1281000
['dropout_1[0][0]']

 flatten_1 (Flatten)            (None, 1000)         0
['Logits[0][0]']

 Predictions (Activation)       (None, 1000)         0
['flatten_1[0][0]']

================================================================================
==================
Total params: 5,507,432
Trainable params: 5,483,032
Non-trainable params: 24,400
--------------------------------------------------------------------------------
------------------
```

[ ]:

```python
base_learning_rate = 1e-4
opt1 = tf.keras.optimizers.Adam(learning_rate=base_learning_rate)
callback_1=tf.keras.callbacks.EarlyStopping(
    monitor='accuracy', min_delta=0, patience=20, verbose=0, mode='auto',
    baseline=None, restore_best_weights=True)

# Call back 2:
callback_2= tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',factor=0.1,
    patience=4,
    verbose=0,
    mode='auto',
    min_delta=0.0001,
    cooldown=0,
    min_lr=0)

callback_list=[callback_1, callback_2]

#compiling our Model for dataset
mobilev3model.compile(optimizer=opt1,
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

# training the model and saving the model components history to history variable
history = mobilev3model.fit(
    train_mobilenet,
    epochs=60,
    validation_data=test_mobilenet,
    class_weight=class_weights,
    callbacks=callback_list)
```

```
Epoch 1/60
```

```
299/299 [==============================] - 48s 61ms/step - loss: 3.0636 -
accuracy: 0.3961 - val_loss: 3.8105 - val_accuracy: 0.1167 - lr: 1.0000e-04
Epoch 2/60
299/299 [==============================] - 17s 57ms/step - loss: 2.6086 -
accuracy: 0.4458 - val_loss: 4.8839 - val_accuracy: 0.1172 - lr: 1.0000e-04
Epoch 3/60
299/299 [==============================] - 17s 57ms/step - loss: 2.2259 -
accuracy: 0.4971 - val_loss: 4.2791 - val_accuracy: 0.1272 - lr: 1.0000e-04
Epoch 4/60
299/299 [==============================] - 17s 57ms/step - loss: 1.9213 -
accuracy: 0.5421 - val_loss: 4.7199 - val_accuracy: 0.1439 - lr: 1.0000e-04
Epoch 5/60
299/299 [==============================] - 17s 57ms/step - loss: 1.6380 -
accuracy: 0.5994 - val_loss: 3.6288 - val_accuracy: 0.2033 - lr: 1.0000e-04
Epoch 6/60
299/299 [==============================] - 17s 58ms/step - loss: 1.4046 -
accuracy: 0.6537 - val_loss: 5.1619 - val_accuracy: 0.1808 - lr: 1.0000e-04
Epoch 7/60
299/299 [==============================] - 17s 57ms/step - loss: 1.3536 -
accuracy: 0.6707 - val_loss: 4.3964 - val_accuracy: 0.1770 - lr: 1.0000e-04
Epoch 8/60
299/299 [==============================] - 17s 57ms/step - loss: 1.2341 -
accuracy: 0.7032 - val_loss: 4.9666 - val_accuracy: 0.1699 - lr: 1.0000e-04
Epoch 9/60
299/299 [==============================] - 17s 57ms/step - loss: 1.0523 -
accuracy: 0.7481 - val_loss: 4.8347 - val_accuracy: 0.1791 - lr: 1.0000e-04
Epoch 10/60
299/299 [==============================] - 17s 58ms/step - loss: 0.7149 -
accuracy: 0.8304 - val_loss: 3.7071 - val_accuracy: 0.2540 - lr: 1.0000e-05
Epoch 11/60
299/299 [==============================] - 17s 57ms/step - loss: 0.5576 -
accuracy: 0.8591 - val_loss: 3.3946 - val_accuracy: 0.2895 - lr: 1.0000e-05
Epoch 12/60
299/299 [==============================] - 17s 57ms/step - loss: 0.4758 -
accuracy: 0.8764 - val_loss: 3.3435 - val_accuracy: 0.3029 - lr: 1.0000e-05
Epoch 13/60
299/299 [==============================] - 17s 58ms/step - loss: 0.4305 -
accuracy: 0.8938 - val_loss: 3.3283 - val_accuracy: 0.3084 - lr: 1.0000e-05
Epoch 14/60
299/299 [==============================] - 17s 57ms/step - loss: 0.3896 -
accuracy: 0.9035 - val_loss: 3.3430 - val_accuracy: 0.3050 - lr: 1.0000e-05
Epoch 15/60
299/299 [==============================] - 17s 57ms/step - loss: 0.3524 -
accuracy: 0.9127 - val_loss: 3.4133 - val_accuracy: 0.3192 - lr: 1.0000e-05
Epoch 16/60
299/299 [==============================] - 17s 57ms/step - loss: 0.3123 -
accuracy: 0.9248 - val_loss: 3.4493 - val_accuracy: 0.3167 - lr: 1.0000e-05
Epoch 17/60
```

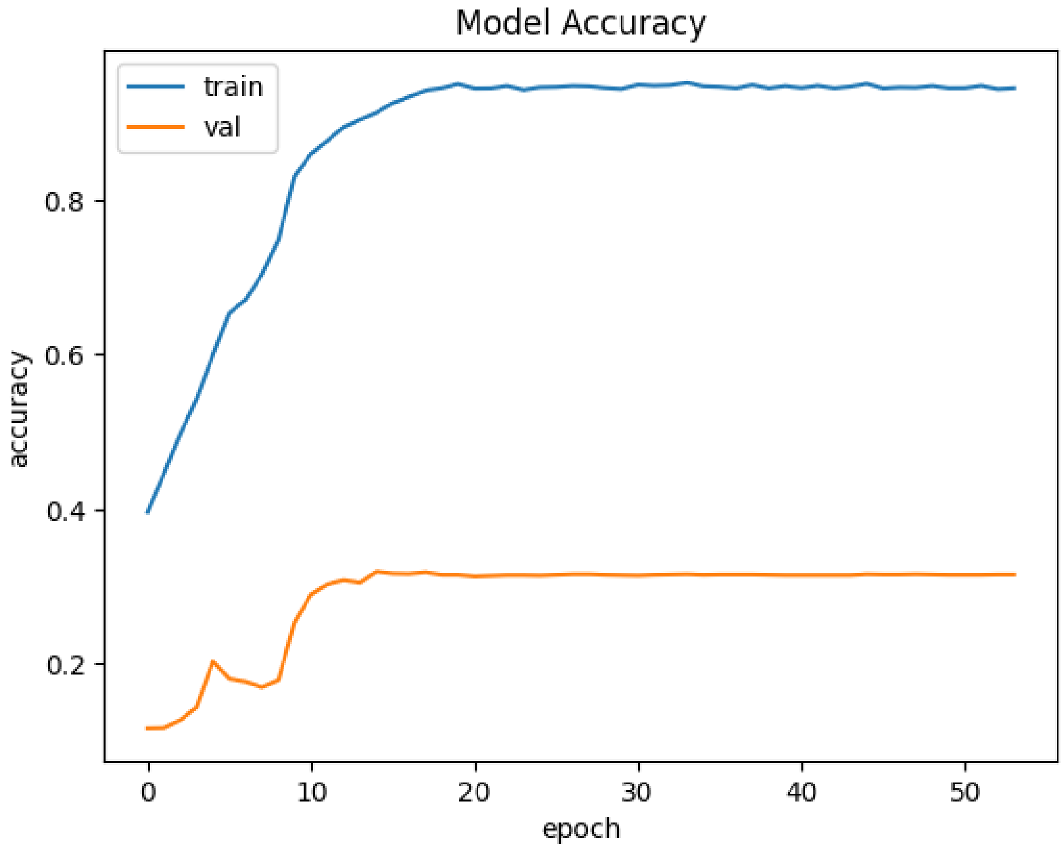
```
299/299 [==============================] - 17s 57ms/step - loss: 0.2777 -
accuracy: 0.9328 - val_loss: 3.5109 - val_accuracy: 0.3163 - lr: 1.0000e-05
Epoch 18/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2536 -
accuracy: 0.9411 - val_loss: 3.4746 - val_accuracy: 0.3184 - lr: 1.0000e-06
Epoch 19/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2478 -
accuracy: 0.9442 - val_loss: 3.4660 - val_accuracy: 0.3151 - lr: 1.0000e-06
Epoch 20/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2212 -
accuracy: 0.9496 - val_loss: 3.4666 - val_accuracy: 0.3151 - lr: 1.0000e-06
Epoch 21/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2357 -
accuracy: 0.9437 - val_loss: 3.4697 - val_accuracy: 0.3130 - lr: 1.0000e-06
Epoch 22/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2352 -
accuracy: 0.9439 - val_loss: 3.4679 - val_accuracy: 0.3138 - lr: 1.0000e-07
Epoch 23/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2268 -
accuracy: 0.9471 - val_loss: 3.4668 - val_accuracy: 0.3146 - lr: 1.0000e-07
Epoch 24/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2346 -
accuracy: 0.9418 - val_loss: 3.4658 - val_accuracy: 0.3146 - lr: 1.0000e-07
Epoch 25/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2360 -
accuracy: 0.9453 - val_loss: 3.4658 - val_accuracy: 0.3142 - lr: 1.0000e-07
Epoch 26/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2368 -
accuracy: 0.9457 - val_loss: 3.4635 - val_accuracy: 0.3151 - lr: 1.0000e-08
Epoch 27/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2382 -
accuracy: 0.9471 - val_loss: 3.4636 - val_accuracy: 0.3159 - lr: 1.0000e-08
Epoch 28/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2294 -
accuracy: 0.9465 - val_loss: 3.4628 - val_accuracy: 0.3159 - lr: 1.0000e-08
Epoch 29/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2378 -
accuracy: 0.9444 - val_loss: 3.4624 - val_accuracy: 0.3151 - lr: 1.0000e-08
Epoch 30/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2464 -
accuracy: 0.9431 - val_loss: 3.4633 - val_accuracy: 0.3146 - lr: 1.0000e-09
Epoch 31/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2345 -
accuracy: 0.9486 - val_loss: 3.4628 - val_accuracy: 0.3142 - lr: 1.0000e-09
Epoch 32/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2243 -
accuracy: 0.9474 - val_loss: 3.4617 - val_accuracy: 0.3151 - lr: 1.0000e-09
Epoch 33/60
```

```
299/299 [==============================] - 17s 58ms/step - loss: 0.2309 -
accuracy: 0.9481 - val_loss: 3.4611 - val_accuracy: 0.3155 - lr: 1.0000e-09
Epoch 34/60
299/299 [==============================] - 18s 58ms/step - loss: 0.2208 -
accuracy: 0.9515 - val_loss: 3.4603 - val_accuracy: 0.3159 - lr: 1.0000e-10
Epoch 35/60
299/299 [==============================] - 18s 58ms/step - loss: 0.2359 -
accuracy: 0.9467 - val_loss: 3.4610 - val_accuracy: 0.3151 - lr: 1.0000e-10
Epoch 36/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2333 -
accuracy: 0.9459 - val_loss: 3.4608 - val_accuracy: 0.3155 - lr: 1.0000e-10
Epoch 37/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2416 -
accuracy: 0.9440 - val_loss: 3.4617 - val_accuracy: 0.3155 - lr: 1.0000e-10
Epoch 38/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2250 -
accuracy: 0.9484 - val_loss: 3.4615 - val_accuracy: 0.3155 - lr: 1.0000e-11
Epoch 39/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2424 -
accuracy: 0.9439 - val_loss: 3.4618 - val_accuracy: 0.3151 - lr: 1.0000e-11
Epoch 40/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2285 -
accuracy: 0.9470 - val_loss: 3.4620 - val_accuracy: 0.3146 - lr: 1.0000e-11
Epoch 41/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2361 -
accuracy: 0.9445 - val_loss: 3.4613 - val_accuracy: 0.3146 - lr: 1.0000e-11
Epoch 42/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2342 -
accuracy: 0.9474 - val_loss: 3.4611 - val_accuracy: 0.3146 - lr: 1.0000e-12
Epoch 43/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2463 -
accuracy: 0.9440 - val_loss: 3.4603 - val_accuracy: 0.3146 - lr: 1.0000e-12
Epoch 44/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2398 -
accuracy: 0.9462 - val_loss: 3.4610 - val_accuracy: 0.3146 - lr: 1.0000e-12
Epoch 45/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2304 -
accuracy: 0.9500 - val_loss: 3.4609 - val_accuracy: 0.3159 - lr: 1.0000e-12
Epoch 46/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2401 -
accuracy: 0.9438 - val_loss: 3.4612 - val_accuracy: 0.3155 - lr: 1.0000e-13
Epoch 47/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2359 -
accuracy: 0.9452 - val_loss: 3.4616 - val_accuracy: 0.3155 - lr: 1.0000e-13
Epoch 48/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2306 -
accuracy: 0.9449 - val_loss: 3.4613 - val_accuracy: 0.3159 - lr: 1.0000e-13
Epoch 49/60
```

```
299/299 [==============================] - 17s 57ms/step - loss: 0.2279 -
accuracy: 0.9471 - val_loss: 3.4606 - val_accuracy: 0.3155 - lr: 1.0000e-13
Epoch 50/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2388 -
accuracy: 0.9441 - val_loss: 3.4612 - val_accuracy: 0.3151 - lr: 1.0000e-14
Epoch 51/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2431 -
accuracy: 0.9442 - val_loss: 3.4606 - val_accuracy: 0.3151 - lr: 1.0000e-14
Epoch 52/60
299/299 [==============================] - 17s 58ms/step - loss: 0.2247 -
accuracy: 0.9473 - val_loss: 3.4605 - val_accuracy: 0.3151 - lr: 1.0000e-14
Epoch 53/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2393 -
accuracy: 0.9430 - val_loss: 3.4609 - val_accuracy: 0.3155 - lr: 1.0000e-14
Epoch 54/60
299/299 [==============================] - 17s 57ms/step - loss: 0.2339 -
accuracy: 0.9441 - val_loss: 3.4620 - val_accuracy: 0.3155 - lr: 1.0000e-15
```

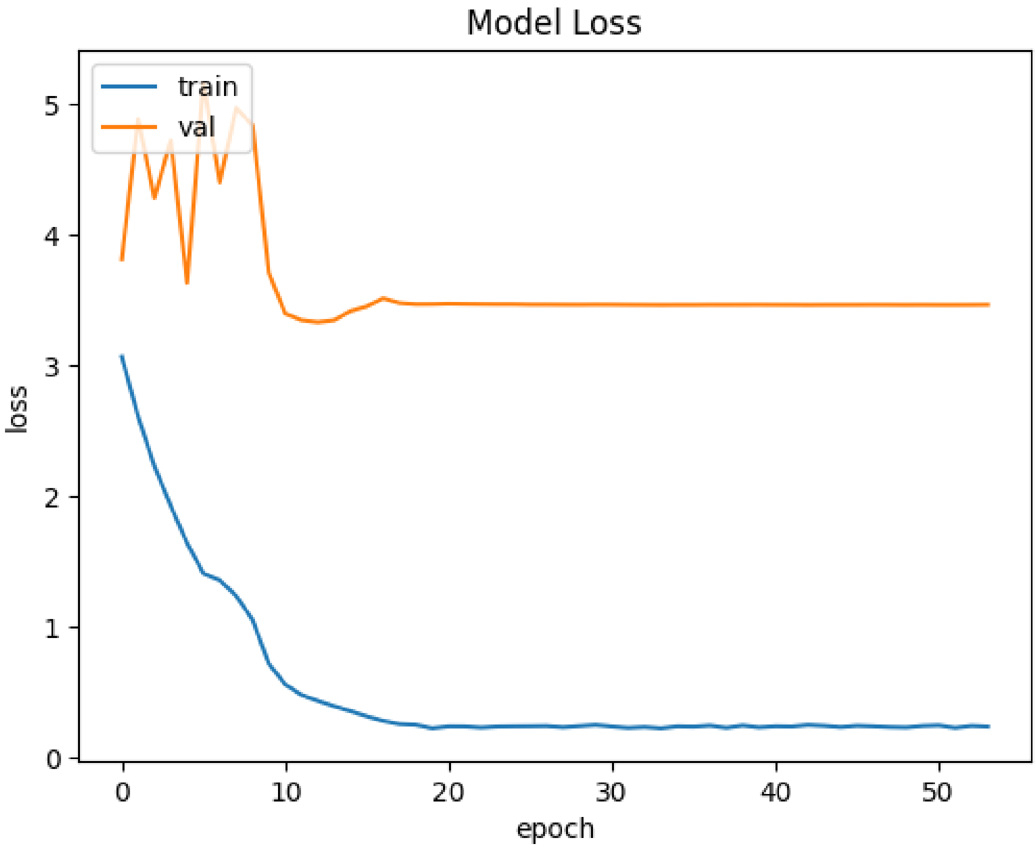
[ ]:

```python
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



[ ]:

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



[ ]:
```
mobilev3model.save(os.path.
  ↪join(SAVE_DIR,'mobileNetV3_FilteredReducedClasses_noPretrain_noTuning.h5'))
```

Iterative Pruning on Most Successful MobilNetV2 and MobileNetV3 Models

[ ]:
```
!pip install -q tensorflow-model-optimization
```

[ ]:
```
import tempfile
import os
import time

import tensorflow as tf
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np

from tensorflow import keras
import tensorflow_model_optimization as tfmot
from keras.callbacks import ModelCheckpoint
```

Define Iterative Pruning Function

```
[ ]: def iterative_pruning(model, initial_sparsity, final_sparsity, begin_step,␣
      ↪end_step, train_data, test_data, class_weights, epochs):
       prune_low_magnitude = tfmot.sparsity.keras.prune_low_magnitude

       # Define model for pruning.
       pruning_params = {
           'pruning_schedule': tfmot.sparsity.keras.
      ↪PolynomialDecay(initial_sparsity=initial_sparsity,
             final_sparsity=final_sparsity, begin_step=begin_step,␣
      ↪end_step=end_step, frequency=50)
       }

       pruned_model = prune_low_magnitude(model, **pruning_params)
       base_learning_rate = 1e-4
       opt1 = tf.keras.optimizers.Adam(learning_rate=base_learning_rate)

       # prune_low_magnitude requires a recompile.
       pruned_model.compile(optimizer=opt1,
                   loss=tf.keras.losses.SparseCategoricalCrossentropy(),
                   metrics=['accuracy'])

       filepath = os.path.join(SAVE_DIR, 'best_pruned_model.epoch{epoch:
      ↪02d}-val_acc{val_accuracy:.2f}.hdf5')
       checkpoint = ModelCheckpoint(filepath=filepath,
                                 monitor='val_accuracy',
                                 verbose=1,
                                 save_best_only=True,
                                 mode='max')
       callbacks = [
         tfmot.sparsity.keras.UpdatePruningStep(),
         checkpoint
       ]

       pruned_model.fit(
           train_data,
           epochs=60,
           validation_data=test_data,
           class_weight=class_weights,
           callbacks=callbacks)

       return pruned_model
```

Attempt Iterative Pruning on Most Successful Model ('mobileNetV2_FilteredReducedClasses_noPretrain_noTuning.h5'; Initially 92.36% Training Accuracy and 49.58 Validation Accuracy)

```
[ ]: model_to_prune = tf.keras.models.load_model(os.path.
      ↪join(SAVE_DIR,'mobileNetV2_FilteredReducedClasses_noPretrain_noTuning.h5'))
     fully_pruned_model = iterative_pruning(model_to_prune, 0, 0.5, 150, 240,␣
      ↪train_mobilenet, test_mobilenet, class_weights, 60)
```

```
Epoch 1/60
299/299 [==============================] - ETA: 0s - loss: 2.8690 - accuracy:
0.6345
Epoch 1: val_accuracy improved from -inf to 0.07280, saving model to /content/dr
ive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch01-val_acc0.07.hdf
5
299/299 [==============================] - 81s 84ms/step - loss: 2.8690 -
accuracy: 0.6345 - val_loss: 4.4822 - val_accuracy: 0.0728
Epoch 2/60
299/299 [==============================] - ETA: 0s - loss: 4.3510 - accuracy:
0.3650
Epoch 2: val_accuracy improved from 0.07280 to 0.19916, saving model to /content
/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch02-val_acc0.20.
hdf5
299/299 [==============================] - 24s 80ms/step - loss: 4.3510 -
accuracy: 0.3650 - val_loss: 2.9680 - val_accuracy: 0.1992
Epoch 3/60
299/299 [==============================] - ETA: 0s - loss: 3.2992 - accuracy:
0.4684
Epoch 3: val_accuracy improved from 0.19916 to 0.25146, saving model to /content
/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch03-val_acc0.25.
hdf5
299/299 [==============================] - 24s 80ms/step - loss: 3.2992 -
accuracy: 0.4684 - val_loss: 2.5498 - val_accuracy: 0.2515
Epoch 4/60
299/299 [==============================] - ETA: 0s - loss: 2.5922 - accuracy:
0.5509
Epoch 4: val_accuracy improved from 0.25146 to 0.32218, saving model to /content
/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch04-val_acc0.32.
hdf5
299/299 [==============================] - 24s 79ms/step - loss: 2.5922 -
accuracy: 0.5509 - val_loss: 2.3622 - val_accuracy: 0.3222
Epoch 5/60
299/299 [==============================] - ETA: 0s - loss: 1.9958 - accuracy:
0.6204
Epoch 5: val_accuracy improved from 0.32218 to 0.36946, saving model to /content
/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch05-val_acc0.37.
hdf5
299/299 [==============================] - 24s 79ms/step - loss: 1.9958 -
accuracy: 0.6204 - val_loss: 2.1605 - val_accuracy: 0.3695
Epoch 6/60
299/299 [==============================] - ETA: 0s - loss: 1.5946 - accuracy:
```

```
0.6870
Epoch 6: val_accuracy did not improve from 0.36946
299/299 [==============================] - 23s 76ms/step - loss: 1.5946 -
accuracy: 0.6870 - val_loss: 2.3622 - val_accuracy: 0.3414
Epoch 7/60
299/299 [==============================] - ETA: 0s - loss: 1.2515 - accuracy:
0.7370
Epoch 7: val_accuracy improved from 0.36946 to 0.37992, saving model to /content
/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch07-val_acc0.38.
hdf5
299/299 [==============================] - 24s 79ms/step - loss: 1.2515 -
accuracy: 0.7370 - val_loss: 2.2412 - val_accuracy: 0.3799
Epoch 8/60
299/299 [==============================] - ETA: 0s - loss: 1.0079 - accuracy:
0.7790
Epoch 8: val_accuracy improved from 0.37992 to 0.38661, saving model to /content
/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch08-val_acc0.39.
hdf5
299/299 [==============================] - 24s 79ms/step - loss: 1.0079 -
accuracy: 0.7790 - val_loss: 2.3380 - val_accuracy: 0.3866
Epoch 9/60
299/299 [==============================] - ETA: 0s - loss: 0.8962 - accuracy:
0.8088
Epoch 9: val_accuracy improved from 0.38661 to 0.38787, saving model to /content
/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch09-val_acc0.39.
hdf5
299/299 [==============================] - 24s 79ms/step - loss: 0.8962 -
accuracy: 0.8088 - val_loss: 2.4176 - val_accuracy: 0.3879
Epoch 10/60
299/299 [==============================] - ETA: 0s - loss: 0.8267 - accuracy:
0.8201
Epoch 10: val_accuracy did not improve from 0.38787
299/299 [==============================] - 23s 76ms/step - loss: 0.8267 -
accuracy: 0.8201 - val_loss: 3.5186 - val_accuracy: 0.3092
Epoch 11/60
299/299 [==============================] - ETA: 0s - loss: 0.6885 - accuracy:
0.8498
Epoch 11: val_accuracy improved from 0.38787 to 0.39205, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch11-val_acc0.39
.hdf5
299/299 [==============================] - 24s 79ms/step - loss: 0.6885 -
accuracy: 0.8498 - val_loss: 2.8400 - val_accuracy: 0.3921
Epoch 12/60
299/299 [==============================] - ETA: 0s - loss: 0.6148 - accuracy:
0.8678
Epoch 12: val_accuracy did not improve from 0.39205
299/299 [==============================] - 23s 77ms/step - loss: 0.6148 -
accuracy: 0.8678 - val_loss: 4.1530 - val_accuracy: 0.3146
```

```
Epoch 13/60
299/299 [==============================] - ETA: 0s - loss: 0.6099 - accuracy:
0.8713
Epoch 13: val_accuracy improved from 0.39205 to 0.40962, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch13-val_acc0.41
.hdf5
299/299 [==============================] - 24s 80ms/step - loss: 0.6099 -
accuracy: 0.8713 - val_loss: 3.0453 - val_accuracy: 0.4096
Epoch 14/60
299/299 [==============================] - ETA: 0s - loss: 0.5738 - accuracy:
0.8785
Epoch 14: val_accuracy improved from 0.40962 to 0.41172, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch14-val_acc0.41
.hdf5
299/299 [==============================] - 24s 80ms/step - loss: 0.5738 -
accuracy: 0.8785 - val_loss: 3.5281 - val_accuracy: 0.4117
Epoch 15/60
299/299 [==============================] - ETA: 0s - loss: 0.5394 - accuracy:
0.8849
Epoch 15: val_accuracy improved from 0.41172 to 0.45021, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch15-val_acc0.45
.hdf5
299/299 [==============================] - 24s 81ms/step - loss: 0.5394 -
accuracy: 0.8849 - val_loss: 2.9398 - val_accuracy: 0.4502
Epoch 16/60
299/299 [==============================] - ETA: 0s - loss: 0.5217 - accuracy:
0.8912
Epoch 16: val_accuracy did not improve from 0.45021
299/299 [==============================] - 23s 77ms/step - loss: 0.5217 -
accuracy: 0.8912 - val_loss: 3.0516 - val_accuracy: 0.3958
Epoch 17/60
299/299 [==============================] - ETA: 0s - loss: 0.4735 - accuracy:
0.9058
Epoch 17: val_accuracy did not improve from 0.45021
299/299 [==============================] - 23s 77ms/step - loss: 0.4735 -
accuracy: 0.9058 - val_loss: 2.7213 - val_accuracy: 0.4439
Epoch 18/60
299/299 [==============================] - ETA: 0s - loss: 0.3739 - accuracy:
0.9231
Epoch 18: val_accuracy did not improve from 0.45021
299/299 [==============================] - 23s 77ms/step - loss: 0.3739 -
accuracy: 0.9231 - val_loss: 4.0699 - val_accuracy: 0.3523
Epoch 19/60
299/299 [==============================] - ETA: 0s - loss: 0.3453 - accuracy:
0.9333
Epoch 19: val_accuracy did not improve from 0.45021
299/299 [==============================] - 23s 77ms/step - loss: 0.3453 -
accuracy: 0.9333 - val_loss: 4.6394 - val_accuracy: 0.3046
```

```
Epoch 20/60
299/299 [==============================] - ETA: 0s - loss: 0.3250 - accuracy:
0.9335
Epoch 20: val_accuracy did not improve from 0.45021
299/299 [==============================] - 23s 76ms/step - loss: 0.3250 -
accuracy: 0.9335 - val_loss: 4.1093 - val_accuracy: 0.3707
Epoch 21/60
299/299 [==============================] - ETA: 0s - loss: 0.3547 - accuracy:
0.9304
Epoch 21: val_accuracy did not improve from 0.45021
299/299 [==============================] - 23s 77ms/step - loss: 0.3547 -
accuracy: 0.9304 - val_loss: 3.6172 - val_accuracy: 0.3866
Epoch 22/60
299/299 [==============================] - ETA: 0s - loss: 0.3725 - accuracy:
0.9253
Epoch 22: val_accuracy improved from 0.45021 to 0.46444, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch22-val_acc0.46
.hdf5
299/299 [==============================] - 24s 80ms/step - loss: 0.3725 -
accuracy: 0.9253 - val_loss: 3.3838 - val_accuracy: 0.4644
Epoch 23/60
299/299 [==============================] - ETA: 0s - loss: 0.3310 - accuracy:
0.9343
Epoch 23: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.3310 -
accuracy: 0.9343 - val_loss: 4.0616 - val_accuracy: 0.3494
Epoch 24/60
299/299 [==============================] - ETA: 0s - loss: 0.2880 - accuracy:
0.9421
Epoch 24: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.2880 -
accuracy: 0.9421 - val_loss: 3.7674 - val_accuracy: 0.4222
Epoch 25/60
299/299 [==============================] - ETA: 0s - loss: 0.2594 - accuracy:
0.9509
Epoch 25: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 76ms/step - loss: 0.2594 -
accuracy: 0.9509 - val_loss: 3.4069 - val_accuracy: 0.4582
Epoch 26/60
299/299 [==============================] - ETA: 0s - loss: 0.2590 - accuracy:
0.9503
Epoch 26: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.2590 -
accuracy: 0.9503 - val_loss: 3.6551 - val_accuracy: 0.4502
Epoch 27/60
299/299 [==============================] - ETA: 0s - loss: 0.2380 - accuracy:
0.9526
Epoch 27: val_accuracy did not improve from 0.46444
```

```
299/299 [==============================] - 23s 77ms/step - loss: 0.2380 -
accuracy: 0.9526 - val_loss: 3.6603 - val_accuracy: 0.4464
Epoch 28/60
299/299 [==============================] - ETA: 0s - loss: 0.2470 - accuracy:
0.9549
Epoch 28: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.2470 -
accuracy: 0.9549 - val_loss: 4.6921 - val_accuracy: 0.4146
Epoch 29/60
299/299 [==============================] - ETA: 0s - loss: 0.2344 - accuracy:
0.9558
Epoch 29: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.2344 -
accuracy: 0.9558 - val_loss: 3.4243 - val_accuracy: 0.4372
Epoch 30/60
299/299 [==============================] - ETA: 0s - loss: 0.2583 - accuracy:
0.9507
Epoch 30: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.2583 -
accuracy: 0.9507 - val_loss: 4.0798 - val_accuracy: 0.4042
Epoch 31/60
299/299 [==============================] - ETA: 0s - loss: 0.3334 - accuracy:
0.9392
Epoch 31: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.3334 -
accuracy: 0.9392 - val_loss: 3.9963 - val_accuracy: 0.4502
Epoch 32/60
299/299 [==============================] - ETA: 0s - loss: 0.2943 - accuracy:
0.9406
Epoch 32: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.2943 -
accuracy: 0.9406 - val_loss: 3.5851 - val_accuracy: 0.4351
Epoch 33/60
299/299 [==============================] - ETA: 0s - loss: 0.2230 - accuracy:
0.9597
Epoch 33: val_accuracy did not improve from 0.46444
299/299 [==============================] - 23s 77ms/step - loss: 0.2230 -
accuracy: 0.9597 - val_loss: 3.7024 - val_accuracy: 0.4460
Epoch 34/60
299/299 [==============================] - ETA: 0s - loss: 0.1750 - accuracy:
0.9686
Epoch 34: val_accuracy improved from 0.46444 to 0.48745, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch34-val_acc0.49
.hdf5
299/299 [==============================] - 24s 80ms/step - loss: 0.1750 -
accuracy: 0.9686 - val_loss: 2.8899 - val_accuracy: 0.4874
Epoch 35/60
299/299 [==============================] - ETA: 0s - loss: 0.1399 - accuracy:
```

```
0.9735
Epoch 35: val_accuracy improved from 0.48745 to 0.48828, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch35-val_acc0.49
.hdf5
299/299 [==============================] - 24s 80ms/step - loss: 0.1399 -
accuracy: 0.9735 - val_loss: 3.0538 - val_accuracy: 0.4883
Epoch 36/60
299/299 [==============================] - ETA: 0s - loss: 0.1860 - accuracy:
0.9662
Epoch 36: val_accuracy did not improve from 0.48828
299/299 [==============================] - 23s 77ms/step - loss: 0.1860 -
accuracy: 0.9662 - val_loss: 3.4499 - val_accuracy: 0.4448
Epoch 37/60
299/299 [==============================] - ETA: 0s - loss: 0.1839 - accuracy:
0.9631
Epoch 37: val_accuracy did not improve from 0.48828
299/299 [==============================] - 23s 77ms/step - loss: 0.1839 -
accuracy: 0.9631 - val_loss: 3.7537 - val_accuracy: 0.4686
Epoch 38/60
299/299 [==============================] - ETA: 0s - loss: 0.1517 - accuracy:
0.9695
Epoch 38: val_accuracy improved from 0.48828 to 0.48870, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch38-val_acc0.49
.hdf5
299/299 [==============================] - 24s 79ms/step - loss: 0.1517 -
accuracy: 0.9695 - val_loss: 3.4282 - val_accuracy: 0.4887
Epoch 39/60
299/299 [==============================] - ETA: 0s - loss: 0.1769 - accuracy:
0.9681
Epoch 39: val_accuracy did not improve from 0.48870
299/299 [==============================] - 23s 77ms/step - loss: 0.1769 -
accuracy: 0.9681 - val_loss: 3.6572 - val_accuracy: 0.4502
Epoch 40/60
299/299 [==============================] - ETA: 0s - loss: 0.2269 - accuracy:
0.9573
Epoch 40: val_accuracy improved from 0.48870 to 0.50084, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch40-val_acc0.50
.hdf5
299/299 [==============================] - 24s 81ms/step - loss: 0.2269 -
accuracy: 0.9573 - val_loss: 3.0037 - val_accuracy: 0.5008
Epoch 41/60
299/299 [==============================] - ETA: 0s - loss: 0.2286 - accuracy:
0.9547
Epoch 41: val_accuracy did not improve from 0.50084
299/299 [==============================] - 24s 80ms/step - loss: 0.2286 -
accuracy: 0.9547 - val_loss: 5.2840 - val_accuracy: 0.3397
Epoch 42/60
299/299 [==============================] - ETA: 0s - loss: 0.2657 - accuracy:
```

```
0.9502
Epoch 42: val_accuracy did not improve from 0.50084
299/299 [==============================] - 24s 79ms/step - loss: 0.2657 -
accuracy: 0.9502 - val_loss: 3.6484 - val_accuracy: 0.4636
Epoch 43/60
299/299 [==============================] - ETA: 0s - loss: 0.2131 - accuracy:
0.9622
Epoch 43: val_accuracy did not improve from 0.50084
299/299 [==============================] - 24s 79ms/step - loss: 0.2131 -
accuracy: 0.9622 - val_loss: 3.2888 - val_accuracy: 0.4481
Epoch 44/60
299/299 [==============================] - ETA: 0s - loss: 0.1754 - accuracy:
0.9690
Epoch 44: val_accuracy did not improve from 0.50084
299/299 [==============================] - 24s 79ms/step - loss: 0.1754 -
accuracy: 0.9690 - val_loss: 3.6128 - val_accuracy: 0.4762
Epoch 45/60
299/299 [==============================] - ETA: 0s - loss: 0.1943 - accuracy:
0.9634
Epoch 45: val_accuracy did not improve from 0.50084
299/299 [==============================] - 24s 79ms/step - loss: 0.1943 -
accuracy: 0.9634 - val_loss: 3.0658 - val_accuracy: 0.4741
Epoch 46/60
299/299 [==============================] - ETA: 0s - loss: 0.1593 - accuracy:
0.9734
Epoch 46: val_accuracy did not improve from 0.50084
299/299 [==============================] - 24s 79ms/step - loss: 0.1593 -
accuracy: 0.9734 - val_loss: 3.8860 - val_accuracy: 0.4757
Epoch 47/60
299/299 [==============================] - ETA: 0s - loss: 0.1254 - accuracy:
0.9751
Epoch 47: val_accuracy improved from 0.50084 to 0.50377, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch47-val_acc0.50
.hdf5
299/299 [==============================] - 24s 82ms/step - loss: 0.1254 -
accuracy: 0.9751 - val_loss: 3.0461 - val_accuracy: 0.5038
Epoch 48/60
299/299 [==============================] - ETA: 0s - loss: 0.0908 - accuracy:
0.9837
Epoch 48: val_accuracy improved from 0.50377 to 0.52845, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch48-val_acc0.53
.hdf5
299/299 [==============================] - 24s 80ms/step - loss: 0.0908 -
accuracy: 0.9837 - val_loss: 3.2661 - val_accuracy: 0.5285
Epoch 49/60
299/299 [==============================] - ETA: 0s - loss: 0.0727 - accuracy:
0.9871
Epoch 49: val_accuracy improved from 0.52845 to 0.54268, saving model to /conten
```

```
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch49-val_acc0.54
.hdf5
299/299 [==============================] - 25s 83ms/step - loss: 0.0727 -
accuracy: 0.9871 - val_loss: 3.2032 - val_accuracy: 0.5427
Epoch 50/60
299/299 [==============================] - ETA: 0s - loss: 0.1119 - accuracy:
0.9789
Epoch 50: val_accuracy did not improve from 0.54268
299/299 [==============================] - 23s 77ms/step - loss: 0.1119 -
accuracy: 0.9789 - val_loss: 3.6675 - val_accuracy: 0.4444
Epoch 51/60
299/299 [==============================] - ETA: 0s - loss: 0.3014 - accuracy:
0.9456
Epoch 51: val_accuracy did not improve from 0.54268
299/299 [==============================] - 23s 78ms/step - loss: 0.3014 -
accuracy: 0.9456 - val_loss: 4.7308 - val_accuracy: 0.4172
Epoch 52/60
299/299 [==============================] - ETA: 0s - loss: 0.2822 - accuracy:
0.9458
Epoch 52: val_accuracy did not improve from 0.54268
299/299 [==============================] - 23s 78ms/step - loss: 0.2822 -
accuracy: 0.9458 - val_loss: 4.3786 - val_accuracy: 0.4121
Epoch 53/60
299/299 [==============================] - ETA: 0s - loss: 0.2130 - accuracy:
0.9608
Epoch 53: val_accuracy did not improve from 0.54268
299/299 [==============================] - 24s 79ms/step - loss: 0.2130 -
accuracy: 0.9608 - val_loss: 3.6218 - val_accuracy: 0.4682
Epoch 54/60
299/299 [==============================] - ETA: 0s - loss: 0.1514 - accuracy:
0.9709
Epoch 54: val_accuracy did not improve from 0.54268
299/299 [==============================] - 24s 79ms/step - loss: 0.1514 -
accuracy: 0.9709 - val_loss: 3.3108 - val_accuracy: 0.4921
Epoch 55/60
299/299 [==============================] - ETA: 0s - loss: 0.1208 - accuracy:
0.9763
Epoch 55: val_accuracy did not improve from 0.54268
299/299 [==============================] - 24s 79ms/step - loss: 0.1208 -
accuracy: 0.9763 - val_loss: 3.8124 - val_accuracy: 0.4883
Epoch 56/60
299/299 [==============================] - ETA: 0s - loss: 0.0800 - accuracy:
0.9846
Epoch 56: val_accuracy did not improve from 0.54268
299/299 [==============================] - 23s 77ms/step - loss: 0.0800 -
accuracy: 0.9846 - val_loss: 3.8132 - val_accuracy: 0.4632
Epoch 57/60
299/299 [==============================] - ETA: 0s - loss: 0.0695 - accuracy:
```

```
0.9871
Epoch 57: val_accuracy improved from 0.54268 to 0.55649, saving model to /conten
t/drive/MyDrive/endg511project/SavedModels/best_pruned_model.epoch57-val_acc0.56
.hdf5
299/299 [==============================] - 24s 80ms/step - loss: 0.0695 -
accuracy: 0.9871 - val_loss: 2.7413 - val_accuracy: 0.5565
Epoch 58/60
299/299 [==============================] - ETA: 0s - loss: 0.0953 - accuracy:
0.9835
Epoch 58: val_accuracy did not improve from 0.55649
299/299 [==============================] - 23s 77ms/step - loss: 0.0953 -
accuracy: 0.9835 - val_loss: 2.7788 - val_accuracy: 0.5117
Epoch 59/60
299/299 [==============================] - ETA: 0s - loss: 0.1737 - accuracy:
0.9685
Epoch 59: val_accuracy did not improve from 0.55649
299/299 [==============================] - 23s 76ms/step - loss: 0.1737 -
accuracy: 0.9685 - val_loss: 3.1701 - val_accuracy: 0.4946
Epoch 60/60
299/299 [==============================] - ETA: 0s - loss: 0.2293 - accuracy:
0.9626
Epoch 60: val_accuracy did not improve from 0.55649
299/299 [==============================] - 23s 77ms/step - loss: 0.2293 -
accuracy: 0.9626 - val_loss: 3.6745 - val_accuracy: 0.4531
```

Summary of Pruned Model

```
[ ]: fully_pruned_model.summary()
```

```
Model: "mobilenetv2_1.00_224"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_3 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 prune_low_magnitude_Conv1 (Pru  (None, 112, 112, 32  1730
['input_3[0][0]']
 neLowMagnitude)                )

 prune_low_magnitude_bn_Conv1 (  (None, 112, 112, 32  129
['prune_low_magnitude_Conv1[0][0]
 PruneLowMagnitude)             )                                '']

 prune_low_magnitude_Conv1_relu  (None, 112, 112, 32  1
['prune_low_magnitude_bn_Conv1[0]
```

```
  (PruneLowMagnitude)           )                               [0]']

 prune_low_magnitude_expanded_c  (None, 112, 112, 32  289
['prune_low_magnitude_Conv1_relu[
 onv_depthwise (PruneLowMagnitu  )                              0][0]']
 de)

 prune_low_magnitude_expanded_c  (None, 112, 112, 32  129
['prune_low_magnitude_expanded_co
 onv_depthwise_BN (PruneLowMagn  )
nv_depthwise[0][0]']
 itude)

 prune_low_magnitude_expanded_c  (None, 112, 112, 32  1
['prune_low_magnitude_expanded_co
 onv_depthwise_relu (PruneLowMa  )
nv_depthwise_BN[0][0]']
 gnitude)

 prune_low_magnitude_expanded_c  (None, 112, 112, 16  1026
['prune_low_magnitude_expanded_co
 onv_project (PruneLowMagnitude  )
nv_depthwise_relu[0][0]']
 )

 prune_low_magnitude_expanded_c  (None, 112, 112, 16  65
['prune_low_magnitude_expanded_co
 onv_project_BN (PruneLowMagnit  )
nv_project[0][0]']
 ude)

 prune_low_magnitude_block_1_ex  (None, 112, 112, 96  3074
['prune_low_magnitude_expanded_co
 pand (PruneLowMagnitude)       )
nv_project_BN[0][0]']

 prune_low_magnitude_block_1_ex  (None, 112, 112, 96  385
['prune_low_magnitude_block_1_exp
 pand_BN (PruneLowMagnitude)    )                               and[0][0]']

 prune_low_magnitude_block_1_ex  (None, 112, 112, 96  1
['prune_low_magnitude_block_1_exp
 pand_relu (PruneLowMagnitude)  )                               and_BN[0][0]']

 prune_low_magnitude_block_1_pa  (None, 113, 113, 96  1
['prune_low_magnitude_block_1_exp
 d (PruneLowMagnitude)          )
and_relu[0][0]']
```

```
 prune_low_magnitude_block_1_de  (None, 56, 56, 96)  865
['prune_low_magnitude_block_1_pad
 pthwise (PruneLowMagnitude)                                     [0][0]']

 prune_low_magnitude_block_1_de  (None, 56, 56, 96)  385
['prune_low_magnitude_block_1_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_1_de  (None, 56, 56, 96)  1
['prune_low_magnitude_block_1_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_1_pr  (None, 56, 56, 24)  4610
['prune_low_magnitude_block_1_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_1_pr  (None, 56, 56, 24)  97
['prune_low_magnitude_block_1_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_2_ex  (None, 56, 56, 144)  6914
['prune_low_magnitude_block_1_pro
 pand (PruneLowMagnitude)                                        ject_BN[0][0]']

 prune_low_magnitude_block_2_ex  (None, 56, 56, 144)  577
['prune_low_magnitude_block_2_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_2_ex  (None, 56, 56, 144)  1
['prune_low_magnitude_block_2_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_2_de  (None, 56, 56, 144)  1297
['prune_low_magnitude_block_2_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_2_de  (None, 56, 56, 144)  577
['prune_low_magnitude_block_2_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_2_de  (None, 56, 56, 144)  1
['prune_low_magnitude_block_2_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_2_pr  (None, 56, 56, 24)  6914
['prune_low_magnitude_block_2_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_2_pr  (None, 56, 56, 24)  97
['prune_low_magnitude_block_2_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_2_ad  (None, 56, 56, 24)  1
['prune_low_magnitude_block_1_pro
 d (PruneLowMagnitude)                                           ject_BN[0][0]',
'prune_low_magnitude_block_2_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_3_ex  (None, 56, 56, 144)  6914
['prune_low_magnitude_block_2_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_3_ex  (None, 56, 56, 144)  577
['prune_low_magnitude_block_3_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_3_ex  (None, 56, 56, 144)  1
['prune_low_magnitude_block_3_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_3_pa  (None, 57, 57, 144)  1
['prune_low_magnitude_block_3_exp
 d (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_3_de  (None, 28, 28, 144)  1297
['prune_low_magnitude_block_3_pad
 pthwise (PruneLowMagnitude)                                     [0][0]']

 prune_low_magnitude_block_3_de  (None, 28, 28, 144)  577
['prune_low_magnitude_block_3_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_3_de  (None, 28, 28, 144)  1
['prune_low_magnitude_block_3_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)
 prune_low_magnitude_block_3_pr  (None, 28, 28, 32)  9218
['prune_low_magnitude_block_3_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_3_pr  (None, 28, 28, 32)  129
['prune_low_magnitude_block_3_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_4_ex  (None, 28, 28, 192)  12290
['prune_low_magnitude_block_3_pro
 pand (PruneLowMagnitude)                                        ject_BN[0][0]']

 prune_low_magnitude_block_4_ex  (None, 28, 28, 192)  769
['prune_low_magnitude_block_4_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_4_ex  (None, 28, 28, 192)  1
['prune_low_magnitude_block_4_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_4_de  (None, 28, 28, 192)  1729
['prune_low_magnitude_block_4_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_4_de  (None, 28, 28, 192)  769
['prune_low_magnitude_block_4_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_4_de  (None, 28, 28, 192)  1
['prune_low_magnitude_block_4_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_4_pr  (None, 28, 28, 32)  12290
['prune_low_magnitude_block_4_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_4_pr  (None, 28, 28, 32)  129
['prune_low_magnitude_block_4_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_4_ad  (None, 28, 28, 32)  1
['prune_low_magnitude_block_3_pro
```

```
 d (PruneLowMagnitude)                                           ject_BN[0][0]',
'prune_low_magnitude_block_4_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_5_ex  (None, 28, 28, 192)  12290
['prune_low_magnitude_block_4_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_5_ex  (None, 28, 28, 192)  769
['prune_low_magnitude_block_5_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_5_ex  (None, 28, 28, 192)  1
['prune_low_magnitude_block_5_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_5_de  (None, 28, 28, 192)  1729
['prune_low_magnitude_block_5_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_5_de  (None, 28, 28, 192)  769
['prune_low_magnitude_block_5_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_5_de  (None, 28, 28, 192)  1
['prune_low_magnitude_block_5_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_5_pr  (None, 28, 28, 32)  12290
['prune_low_magnitude_block_5_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_5_pr  (None, 28, 28, 32)  129
['prune_low_magnitude_block_5_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_5_ad  (None, 28, 28, 32)  1
['prune_low_magnitude_block_4_add
 d (PruneLowMagnitude)                                           [0][0]',
'prune_low_magnitude_block_5_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_6_ex  (None, 28, 28, 192)  12290
['prune_low_magnitude_block_5_add
```

```
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_6_ex  (None, 28, 28, 192)  769
['prune_low_magnitude_block_6_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_6_ex  (None, 28, 28, 192)  1
['prune_low_magnitude_block_6_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_6_pa  (None, 29, 29, 192)  1
['prune_low_magnitude_block_6_exp
 d (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_6_de  (None, 14, 14, 192)  1729
['prune_low_magnitude_block_6_pad
 pthwise (PruneLowMagnitude)                                     [0][0]']

 prune_low_magnitude_block_6_de  (None, 14, 14, 192)  769
['prune_low_magnitude_block_6_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_6_de  (None, 14, 14, 192)  1
['prune_low_magnitude_block_6_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_6_pr  (None, 14, 14, 64)  24578
['prune_low_magnitude_block_6_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_6_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_6_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_7_ex  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_6_pro
 pand (PruneLowMagnitude)                                        ject_BN[0][0]']

 prune_low_magnitude_block_7_ex  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_7_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_7_ex  (None, 14, 14, 384)  1
['prune_low_magnitude_block_7_exp
```

```
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_7_de  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_7_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_7_de  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_7_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_7_de  (None, 14, 14, 384)  1
['prune_low_magnitude_block_7_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_7_pr  (None, 14, 14, 64)  49154
['prune_low_magnitude_block_7_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_7_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_7_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_7_ad  (None, 14, 14, 64)  1
['prune_low_magnitude_block_6_pro
 d (PruneLowMagnitude)                                           ject_BN[0][0]',
'prune_low_magnitude_block_7_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_8_ex  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_7_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_8_ex  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_8_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_8_ex  (None, 14, 14, 384)  1
['prune_low_magnitude_block_8_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_8_de  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_8_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']
```

```
 prune_low_magnitude_block_8_de  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_8_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_8_de  (None, 14, 14, 384)  1
['prune_low_magnitude_block_8_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_8_pr  (None, 14, 14, 64)  49154
['prune_low_magnitude_block_8_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_8_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_8_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_8_ad  (None, 14, 14, 64)  1
['prune_low_magnitude_block_7_add
 d (PruneLowMagnitude)                                           [0][0]',
'prune_low_magnitude_block_8_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_9_ex  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_8_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_9_ex  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_9_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_9_ex  (None, 14, 14, 384)  1
['prune_low_magnitude_block_9_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_9_de  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_9_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_9_de  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_9_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_9_de  (None, 14, 14, 384)  1
```

```
['prune_low_magnitude_block_9_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_9_pr  (None, 14, 14, 64)  49154
['prune_low_magnitude_block_9_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_9_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_9_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_9_ad  (None, 14, 14, 64)  1
['prune_low_magnitude_block_8_add
 d (PruneLowMagnitude)                                           [0][0]',
'prune_low_magnitude_block_9_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_10_e  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_9_add
 xpand (PruneLowMagnitude)                                       [0][0]']

 prune_low_magnitude_block_10_e  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_10_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_10_e  (None, 14, 14, 384)  1
['prune_low_magnitude_block_10_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_10_d  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_10_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_10_d  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_10_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_10_d  (None, 14, 14, 384)  1
['prune_low_magnitude_block_10_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)
 prune_low_magnitude_block_10_p  (None, 14, 14, 96)  73730
['prune_low_magnitude_block_10_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_10_p  (None, 14, 14, 96)  385
['prune_low_magnitude_block_10_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_11_e  (None, 14, 14, 576)  110594
['prune_low_magnitude_block_10_pr
 xpand (PruneLowMagnitude)
oject_BN[0][0]']

 prune_low_magnitude_block_11_e  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_11_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_11_e  (None, 14, 14, 576)  1
['prune_low_magnitude_block_11_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_11_d  (None, 14, 14, 576)  5185
['prune_low_magnitude_block_11_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_11_d  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_11_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_11_d  (None, 14, 14, 576)  1
['prune_low_magnitude_block_11_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_11_p  (None, 14, 14, 96)  110594
['prune_low_magnitude_block_11_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_11_p  (None, 14, 14, 96)  385
['prune_low_magnitude_block_11_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_11_a  (None, 14, 14, 96)  1
```

```
['prune_low_magnitude_block_10_pr
 dd (PruneLowMagnitude)
oject_BN[0][0]',
'prune_low_magnitude_block_11_pr
oject_BN[0][0]']

 prune_low_magnitude_block_12_e  (None, 14, 14, 576)  110594
['prune_low_magnitude_block_11_ad
 xpand (PruneLowMagnitude)                                       d[0][0]']

 prune_low_magnitude_block_12_e  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_12_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_12_e  (None, 14, 14, 576)  1
['prune_low_magnitude_block_12_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_12_d  (None, 14, 14, 576)  5185
['prune_low_magnitude_block_12_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_12_d  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_12_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_12_d  (None, 14, 14, 576)  1
['prune_low_magnitude_block_12_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_12_p  (None, 14, 14, 96)  110594
['prune_low_magnitude_block_12_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_12_p  (None, 14, 14, 96)  385
['prune_low_magnitude_block_12_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_12_a  (None, 14, 14, 96)  1
['prune_low_magnitude_block_11_ad
 dd (PruneLowMagnitude)                                          d[0][0]',
'prune_low_magnitude_block_12_pr
oject_BN[0][0]']
```

```
 prune_low_magnitude_block_13_e  (None, 14, 14, 576)  110594
['prune_low_magnitude_block_12_ad
 xpand (PruneLowMagnitude)                                       d[0][0]']

 prune_low_magnitude_block_13_e  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_13_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_13_e  (None, 14, 14, 576)  1
['prune_low_magnitude_block_13_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_13_p  (None, 15, 15, 576)  1
['prune_low_magnitude_block_13_ex
 ad (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_13_d  (None, 7, 7, 576)   5185
['prune_low_magnitude_block_13_pa
 epthwise (PruneLowMagnitude)                                    d[0][0]']

 prune_low_magnitude_block_13_d  (None, 7, 7, 576)   2305
['prune_low_magnitude_block_13_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_13_d  (None, 7, 7, 576)   1
['prune_low_magnitude_block_13_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_13_p  (None, 7, 7, 160)   184322
['prune_low_magnitude_block_13_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_13_p  (None, 7, 7, 160)   641
['prune_low_magnitude_block_13_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_14_e  (None, 7, 7, 960)   307202
['prune_low_magnitude_block_13_pr
 xpand (PruneLowMagnitude)
oject_BN[0][0]']

 prune_low_magnitude_block_14_e  (None, 7, 7, 960)   3841
```

```
['prune_low_magnitude_block_14_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_14_e  (None, 7, 7, 960)   1
['prune_low_magnitude_block_14_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_14_d  (None, 7, 7, 960)   8641
['prune_low_magnitude_block_14_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_14_d  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_14_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_14_d  (None, 7, 7, 960)   1
['prune_low_magnitude_block_14_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_14_p  (None, 7, 7, 160)   307202
['prune_low_magnitude_block_14_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_14_p  (None, 7, 7, 160)   641
['prune_low_magnitude_block_14_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_14_a  (None, 7, 7, 160)   1
['prune_low_magnitude_block_13_pr
 dd (PruneLowMagnitude)
oject_BN[0][0]',
'prune_low_magnitude_block_14_pr
oject_BN[0][0]']

 prune_low_magnitude_block_15_e  (None, 7, 7, 960)   307202
['prune_low_magnitude_block_14_ad
 xpand (PruneLowMagnitude)                                       d[0][0]']

 prune_low_magnitude_block_15_e  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_15_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_15_e  (None, 7, 7, 960)   1
```

```
['prune_low_magnitude_block_15_ex
 xpand_relu (PruneLowMagnitude)                                 pand_BN[0][0]']

 prune_low_magnitude_block_15_d  (None, 7, 7, 960)   8641
['prune_low_magnitude_block_15_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_15_d  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_15_de
 epthwise_BN (PruneLowMagnitude                                 pthwise[0][0]']
 )

 prune_low_magnitude_block_15_d  (None, 7, 7, 960)   1
['prune_low_magnitude_block_15_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_15_p  (None, 7, 7, 160)   307202
['prune_low_magnitude_block_15_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_15_p  (None, 7, 7, 160)   641
['prune_low_magnitude_block_15_pr
 roject_BN (PruneLowMagnitude)                                  oject[0][0]']

 prune_low_magnitude_block_15_a  (None, 7, 7, 160)   1
['prune_low_magnitude_block_14_ad
 dd (PruneLowMagnitude)                                         d[0][0]',
'prune_low_magnitude_block_15_pr
oject_BN[0][0]']

 prune_low_magnitude_block_16_e  (None, 7, 7, 960)   307202
['prune_low_magnitude_block_15_ad
 xpand (PruneLowMagnitude)                                      d[0][0]']

 prune_low_magnitude_block_16_e  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_16_ex
 xpand_BN (PruneLowMagnitude)                                   pand[0][0]']

 prune_low_magnitude_block_16_e  (None, 7, 7, 960)   1
['prune_low_magnitude_block_16_ex
 xpand_relu (PruneLowMagnitude)                                 pand_BN[0][0]']

 prune_low_magnitude_block_16_d  (None, 7, 7, 960)   8641
['prune_low_magnitude_block_16_ex
```

```
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_16_d  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_16_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_16_d  (None, 7, 7, 960)   1
['prune_low_magnitude_block_16_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_16_p  (None, 7, 7, 320)   614402
['prune_low_magnitude_block_16_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_16_p  (None, 7, 7, 320)   1281
['prune_low_magnitude_block_16_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_Conv_1 (Pr  (None, 7, 7, 1280)  819202
['prune_low_magnitude_block_16_pr
 uneLowMagnitude)
oject_BN[0][0]']

 prune_low_magnitude_Conv_1_bn   (None, 7, 7, 1280)  5121
['prune_low_magnitude_Conv_1[0][0
 (PruneLowMagnitude)                                             ]']

 prune_low_magnitude_out_relu (  (None, 7, 7, 1280)  1
['prune_low_magnitude_Conv_1_bn[0
 PruneLowMagnitude)                                              ][0]']

 prune_low_magnitude_global_ave  (None, 1280)        1
['prune_low_magnitude_out_relu[0]
 rage_pooling2d_2 (PruneLowMagn                                  [0]']
 itude)

 prune_low_magnitude_prediction  (None, 12)          30734
['prune_low_magnitude_global_aver
 s (PruneLowMagnitude)
age_pooling2d_2[0][0]']

================================================================================
==================
Total params: 4,414,443
Trainable params: 2,239,244
Non-trainable params: 2,175,199
________________________________________________________________________________
__________________
```

Apply Strip Pruning to Pruned Model

[ ]:
```python
# Strip pruning wrappers
stripped_pruned_model = tfmot.sparsity.keras.strip_pruning(fully_pruned_model)
```

[ ]:
```python
stripped_pruned_model.save(os.path.join(SAVE_DIR,'stripped_pruned_model.h5'))
```

```
WARNING:tensorflow:Compiled the loaded model, but the compiled metrics have yet
to be built. `model.compile_metrics` will be empty until you train or evaluate
the model.
```

Compare Model Sizes

[ ]:
```python
# Evaluate Model Size
def get_gzipped_model_size(file):
  # Returns size of gzipped model, in bytes.
  import os
  import zipfile

  _, zipped_file = tempfile.mkstemp('.zip')
  with zipfile.ZipFile(zipped_file, 'w', compression=zipfile.ZIP_DEFLATED) as f:
    f.write(file)

  return os.path.getsize(zipped_file)
```

[ ]:
```python
print(f"Unpruned Model Size: {get_gzipped_model_size(os.path.
 ↪join(SAVE_DIR,'mobileNetV2_FilteredReducedClasses_noPretrain_noTuning.
 ↪h5'))}")
print(f"Pruned Model Size: {get_gzipped_model_size(os.path.
 ↪join(SAVE_DIR,'stripped_pruned_model.h5'))}")
```

```
Unpruned Model Size: 24799951
Pruned Model Size: 5688732
```

Attempt to load best performing model failed, turned out to be a known issue in Keras.

[ ]:
```python
model_to_strip_prune = tf.keras.models.load_model(os.path.
 ↪join(SAVE_DIR,'best_pruned_model.epoch57-val_acc0.56.hdf5'))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-99-6c28fed3545a> in <cell line: 1>()
----> 1 model_to_strip_prune = tf.keras.models.load_model(os.path.
 ↪join(SAVE_DIR,'best_pruned_model.epoch57-val_acc0.56.hdf5'))
```

```
/usr/local/lib/python3.9/dist-packages/keras/saving/saving_api.py in␣
 ↪load_model(filepath, custom_objects, compile, safe_mode, **kwargs)
    210
    211     # Legacy case.
--> 212     return legacy_sm_saving_lib.load_model(
    213         filepath, custom_objects=custom_objects, compile=compile,␣
 ↪**kwargs
    214     )

/usr/local/lib/python3.9/dist-packages/keras/utils/traceback_utils.py in␣
 ↪error_handler(*args, **kwargs)
     68             # To get the full stack trace, call:
     69             # `tf.debugging.disable_traceback_filtering()`
---> 70             raise e.with_traceback(filtered_tb) from None
     71         finally:
     72             del filtered_tb

/usr/local/lib/python3.9/dist-packages/keras/saving/legacy/serialization.py in␣
 ↪class_and_config_for_serialized_keras_object(config, module_objects,␣
 ↪custom_objects, printable_module_name)
    366     )
    367     if cls is None:
--> 368         raise ValueError(
    369             f"Unknown {printable_module_name}: '{class_name}'. "
    370             "Please ensure you are using a `keras.utils.
 ↪custom_object_scope` "

ValueError: Unknown layer: 'PruneLowMagnitude'. Please ensure you are using a␣
 ↪`keras.utils.custom_object_scope` and that this object is included in the␣
 ↪scope. See https://www.tensorflow.org/guide/keras/
 ↪save_and_serialize#registering_the_custom_object for details.
```

[ ]:

```
def print_model_weights_sparsity(model):

    for layer in model.layers:
        if isinstance(layer, tf.keras.layers.Wrapper):
            weights = layer.trainable_weights
        else:
            weights = layer.weights
        for weight in weights:
            if "kernel" not in weight.name or "centroid" in weight.name:
                continue
            weight_size = weight.numpy().size
            zero_num = np.count_nonzero(weight == 0)
            print(
                f"{weight.name}: {zero_num/weight_size:.2%} sparsity ",
                f"({zero_num}/{weight_size})",
            )
```

Printout Sparsity

[ ]:
```
print_model_weights_sparsity(stripped_pruned_model)
```

```
Conv1/kernel:0: 45.60% sparsity  (394/864)
expanded_conv_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/288)
expanded_conv_project/kernel:0: 45.70% sparsity  (234/512)
block_1_expand/kernel:0: 45.64% sparsity  (701/1536)
block_1_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/864)
block_1_project/kernel:0: 45.62% sparsity  (1051/2304)
block_2_expand/kernel:0: 45.60% sparsity  (1576/3456)
block_2_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1296)
block_2_project/kernel:0: 45.60% sparsity  (1576/3456)
block_3_expand/kernel:0: 45.60% sparsity  (1576/3456)
block_3_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1296)
block_3_project/kernel:0: 45.62% sparsity  (2102/4608)
block_4_expand/kernel:0: 45.61% sparsity  (2802/6144)
block_4_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1728)
block_4_project/kernel:0: 45.61% sparsity  (2802/6144)
block_5_expand/kernel:0: 45.61% sparsity  (2802/6144)
block_5_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1728)
block_5_project/kernel:0: 45.61% sparsity  (2802/6144)
block_6_expand/kernel:0: 45.61% sparsity  (2802/6144)
block_6_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1728)
block_6_project/kernel:0: 45.61% sparsity  (5605/12288)
block_7_expand/kernel:0: 45.61% sparsity  (11209/24576)
block_7_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_7_project/kernel:0: 45.61% sparsity  (11209/24576)
block_8_expand/kernel:0: 45.61% sparsity  (11209/24576)
block_8_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_8_project/kernel:0: 45.61% sparsity  (11209/24576)
block_9_expand/kernel:0: 45.61% sparsity  (11209/24576)
block_9_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_9_project/kernel:0: 45.61% sparsity  (11209/24576)
block_10_expand/kernel:0: 45.61% sparsity  (11209/24576)
block_10_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_10_project/kernel:0: 45.61% sparsity  (16814/36864)
block_11_expand/kernel:0: 45.61% sparsity  (25221/55296)
block_11_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/5184)
block_11_project/kernel:0: 45.61% sparsity  (25221/55296)
block_12_expand/kernel:0: 45.61% sparsity  (25221/55296)
block_12_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/5184)
block_12_project/kernel:0: 45.61% sparsity  (25221/55296)
block_13_expand/kernel:0: 45.61% sparsity  (25221/55296)
block_13_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/5184)
```

```
block_13_project/kernel:0: 45.61% sparsity  (42035/92160)
block_14_expand/kernel:0: 45.61% sparsity  (70058/153600)
block_14_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/8640)
block_14_project/kernel:0: 45.61% sparsity  (70058/153600)
block_15_expand/kernel:0: 45.61% sparsity  (70058/153600)
block_15_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/8640)
block_15_project/kernel:0: 45.61% sparsity  (70058/153600)
block_16_expand/kernel:0: 45.61% sparsity  (70058/153600)
block_16_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/8640)
block_16_project/kernel:0: 45.61% sparsity  (140115/307200)
Conv_1/kernel:0: 45.61% sparsity  (186820/409600)
predictions/kernel:0: 45.61% sparsity  (7006/15360)
```

One Shot Sparsity

[ ]:
```
prune_low_magnitude = tfmot.sparsity.keras.prune_low_magnitude
## Print weights before and after

# Define model for pruning. The 0.5 is the target sparsity (50%)
pruning_params = {
    'pruning_schedule': tfmot.sparsity.keras.ConstantSparsity(0.5,␣
 ↪begin_step=0, frequency=100)
}

pruned_model_one_shot = prune_low_magnitude(model_to_prune, **pruning_params)

# prune_low_magnitude requires a recompile.
optimizer = tf.keras.optimizers.Adam(learning_rate=1e-5)
pruned_model_one_shot.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(),
              metrics=['accuracy'])

pruned_model_one_shot.summary()
```

```
Model: "mobilenetv2_1.00_224"
________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
================================================================================
==================
 input_3 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 prune_low_magnitude_Conv1 (Pru  (None, 112, 112, 32  1730
['input_3[0][0]']
 neLowMagnitude)                )

 prune_low_magnitude_bn_Conv1 (  (None, 112, 112, 32  129
['prune_low_magnitude_Conv1[0][0]
 PruneLowMagnitude)             )                                '[']

 prune_low_magnitude_Conv1_relu  (None, 112, 112, 32  1
['prune_low_magnitude_bn_Conv1[0]
  (PruneLowMagnitude)           )                                [0]']

 prune_low_magnitude_expanded_c  (None, 112, 112, 32  289
['prune_low_magnitude_Conv1_relu[
 onv_depthwise (PruneLowMagnitu  )                               0][0]']
 de)

 prune_low_magnitude_expanded_c  (None, 112, 112, 32  129
['prune_low_magnitude_expanded_co
 onv_depthwise_BN (PruneLowMagn  )
nv_depthwise[0][0]']
 itude)

 prune_low_magnitude_expanded_c  (None, 112, 112, 32  1
['prune_low_magnitude_expanded_co
 onv_depthwise_relu (PruneLowMa  )
nv_depthwise_BN[0][0]']
 gnitude)

 prune_low_magnitude_expanded_c  (None, 112, 112, 16  1026
['prune_low_magnitude_expanded_co
 onv_project (PruneLowMagnitude  )
nv_depthwise_relu[0][0]']
 )

 prune_low_magnitude_expanded_c  (None, 112, 112, 16  65
['prune_low_magnitude_expanded_co
 onv_project_BN (PruneLowMagnit  )
nv_project[0][0]']
 ude)

 prune_low_magnitude_block_1_ex  (None, 112, 112, 96  3074
['prune_low_magnitude_expanded_co
 pand (PruneLowMagnitude)       )
nv_project_BN[0][0]']

 prune_low_magnitude_block_1_ex  (None, 112, 112, 96  385
['prune_low_magnitude_block_1_exp
 pand_BN (PruneLowMagnitude)    )                                and[0][0]']

 prune_low_magnitude_block_1_ex  (None, 112, 112, 96  1
['prune_low_magnitude_block_1_exp
 pand_relu (PruneLowMagnitude)  )                                and_BN[0][0]']
 prune_low_magnitude_block_1_pa  (None, 113, 113, 96  1
['prune_low_magnitude_block_1_exp
 d (PruneLowMagnitude)          )
and_relu[0][0]']

 prune_low_magnitude_block_1_de  (None, 56, 56, 96)   865
['prune_low_magnitude_block_1_pad
 pthwise (PruneLowMagnitude)                                     [0][0]']

 prune_low_magnitude_block_1_de  (None, 56, 56, 96)   385
['prune_low_magnitude_block_1_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_1_de  (None, 56, 56, 96)  1
['prune_low_magnitude_block_1_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_1_pr  (None, 56, 56, 24)  4610
['prune_low_magnitude_block_1_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_1_pr  (None, 56, 56, 24)  97
['prune_low_magnitude_block_1_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_2_ex  (None, 56, 56, 144)  6914
['prune_low_magnitude_block_1_pro
 pand (PruneLowMagnitude)                                        ject_BN[0][0]']

 prune_low_magnitude_block_2_ex  (None, 56, 56, 144)  577
['prune_low_magnitude_block_2_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_2_ex  (None, 56, 56, 144)  1
['prune_low_magnitude_block_2_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_2_de  (None, 56, 56, 144)  1297
['prune_low_magnitude_block_2_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_2_de  (None, 56, 56, 144)  577
['prune_low_magnitude_block_2_dep
```

```
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_2_de  (None, 56, 56, 144)  1
['prune_low_magnitude_block_2_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_2_pr  (None, 56, 56, 24)  6914
['prune_low_magnitude_block_2_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_2_pr  (None, 56, 56, 24)  97
['prune_low_magnitude_block_2_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_2_ad  (None, 56, 56, 24)  1
['prune_low_magnitude_block_1_pro
 d (PruneLowMagnitude)                                           ject_BN[0][0]',
'prune_low_magnitude_block_2_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_3_ex  (None, 56, 56, 144)  6914
['prune_low_magnitude_block_2_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_3_ex  (None, 56, 56, 144)  577
['prune_low_magnitude_block_3_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_3_ex  (None, 56, 56, 144)  1
['prune_low_magnitude_block_3_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_3_pa  (None, 57, 57, 144)  1
['prune_low_magnitude_block_3_exp
 d (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_3_de  (None, 28, 28, 144)  1297
['prune_low_magnitude_block_3_pad
 pthwise (PruneLowMagnitude)                                     [0][0]']

 prune_low_magnitude_block_3_de  (None, 28, 28, 144)  577
['prune_low_magnitude_block_3_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']
 prune_low_magnitude_block_3_de  (None, 28, 28, 144)  1
['prune_low_magnitude_block_3_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_3_pr  (None, 28, 28, 32)  9218
['prune_low_magnitude_block_3_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_3_pr  (None, 28, 28, 32)  129
['prune_low_magnitude_block_3_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_4_ex  (None, 28, 28, 192)  12290
['prune_low_magnitude_block_3_pro
 pand (PruneLowMagnitude)                                        ject_BN[0][0]']

 prune_low_magnitude_block_4_ex  (None, 28, 28, 192)  769
['prune_low_magnitude_block_4_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_4_ex  (None, 28, 28, 192)  1
['prune_low_magnitude_block_4_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_4_de  (None, 28, 28, 192)  1729
['prune_low_magnitude_block_4_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_4_de  (None, 28, 28, 192)  769
['prune_low_magnitude_block_4_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_4_de  (None, 28, 28, 192)  1
['prune_low_magnitude_block_4_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_4_pr  (None, 28, 28, 32)  12290
['prune_low_magnitude_block_4_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_4_pr  (None, 28, 28, 32)  129
```

```
['prune_low_magnitude_block_4_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_4_ad  (None, 28, 28, 32)  1
['prune_low_magnitude_block_3_pro
 d (PruneLowMagnitude)                                           ject_BN[0][0]',
'prune_low_magnitude_block_4_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_5_ex  (None, 28, 28, 192)  12290
['prune_low_magnitude_block_4_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_5_ex  (None, 28, 28, 192)  769
['prune_low_magnitude_block_5_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_5_ex  (None, 28, 28, 192)  1
['prune_low_magnitude_block_5_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_5_de  (None, 28, 28, 192)  1729
['prune_low_magnitude_block_5_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_5_de  (None, 28, 28, 192)  769
['prune_low_magnitude_block_5_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_5_de  (None, 28, 28, 192)  1
['prune_low_magnitude_block_5_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_5_pr  (None, 28, 28, 32)  12290
['prune_low_magnitude_block_5_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_5_pr  (None, 28, 28, 32)  129
['prune_low_magnitude_block_5_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_5_ad  (None, 28, 28, 32)  1
['prune_low_magnitude_block_4_add
 d (PruneLowMagnitude)                                           [0][0]',
'prune_low_magnitude_block_5_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_6_ex  (None, 28, 28, 192)  12290
['prune_low_magnitude_block_5_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_6_ex  (None, 28, 28, 192)  769
['prune_low_magnitude_block_6_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_6_ex  (None, 28, 28, 192)  1
['prune_low_magnitude_block_6_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_6_pa  (None, 29, 29, 192)  1
['prune_low_magnitude_block_6_exp
 d (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_6_de  (None, 14, 14, 192)  1729
['prune_low_magnitude_block_6_pad
 pthwise (PruneLowMagnitude)                                     [0][0]']

 prune_low_magnitude_block_6_de  (None, 14, 14, 192)  769
['prune_low_magnitude_block_6_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_6_de  (None, 14, 14, 192)  1
['prune_low_magnitude_block_6_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_6_pr  (None, 14, 14, 64)  24578
['prune_low_magnitude_block_6_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_6_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_6_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_7_ex  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_6_pro
 pand (PruneLowMagnitude)                                        ject_BN[0][0]']

 prune_low_magnitude_block_7_ex  (None, 14, 14, 384)  1537
```

```
['prune_low_magnitude_block_7_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_7_ex  (None, 14, 14, 384)  1
['prune_low_magnitude_block_7_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_7_de  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_7_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_7_de  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_7_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_7_de  (None, 14, 14, 384)  1
['prune_low_magnitude_block_7_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_7_pr  (None, 14, 14, 64)  49154
['prune_low_magnitude_block_7_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_7_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_7_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_7_ad  (None, 14, 14, 64)  1
['prune_low_magnitude_block_6_pro
 d (PruneLowMagnitude)                                           ject_BN[0][0]',
'prune_low_magnitude_block_7_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_8_ex  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_7_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_8_ex  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_8_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_8_ex  (None, 14, 14, 384)  1
['prune_low_magnitude_block_8_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']
 prune_low_magnitude_block_8_de  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_8_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']

 prune_low_magnitude_block_8_de  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_8_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_8_de  (None, 14, 14, 384)  1
['prune_low_magnitude_block_8_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_8_pr  (None, 14, 14, 64)  49154
['prune_low_magnitude_block_8_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_8_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_8_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_8_ad  (None, 14, 14, 64)  1
['prune_low_magnitude_block_7_add
 d (PruneLowMagnitude)                                           [0][0]',
'prune_low_magnitude_block_8_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_9_ex  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_8_add
 pand (PruneLowMagnitude)                                        [0][0]']

 prune_low_magnitude_block_9_ex  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_9_exp
 pand_BN (PruneLowMagnitude)                                     and[0][0]']

 prune_low_magnitude_block_9_ex  (None, 14, 14, 384)  1
['prune_low_magnitude_block_9_exp
 pand_relu (PruneLowMagnitude)                                   and_BN[0][0]']

 prune_low_magnitude_block_9_de  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_9_exp
 pthwise (PruneLowMagnitude)
and_relu[0][0]']
```

```
 prune_low_magnitude_block_9_de  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_9_dep
 pthwise_BN (PruneLowMagnitude)                                  thwise[0][0]']

 prune_low_magnitude_block_9_de  (None, 14, 14, 384)  1
['prune_low_magnitude_block_9_dep
 pthwise_relu (PruneLowMagnitud
thwise_BN[0][0]']
 e)

 prune_low_magnitude_block_9_pr  (None, 14, 14, 64)  49154
['prune_low_magnitude_block_9_dep
 oject (PruneLowMagnitude)
thwise_relu[0][0]']

 prune_low_magnitude_block_9_pr  (None, 14, 14, 64)  257
['prune_low_magnitude_block_9_pro
 oject_BN (PruneLowMagnitude)                                    ject[0][0]']

 prune_low_magnitude_block_9_ad  (None, 14, 14, 64)  1
['prune_low_magnitude_block_8_add
 d (PruneLowMagnitude)                                           [0][0]',
'prune_low_magnitude_block_9_pro
                                                                 ject_BN[0][0]']

 prune_low_magnitude_block_10_e  (None, 14, 14, 384)  49154
['prune_low_magnitude_block_9_add
 xpand (PruneLowMagnitude)                                       [0][0]']

 prune_low_magnitude_block_10_e  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_10_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_10_e  (None, 14, 14, 384)  1
['prune_low_magnitude_block_10_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_10_d  (None, 14, 14, 384)  3457
['prune_low_magnitude_block_10_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_10_d  (None, 14, 14, 384)  1537
['prune_low_magnitude_block_10_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_10_d  (None, 14, 14, 384)  1
```

```
['prune_low_magnitude_block_10_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_10_p  (None, 14, 14, 96)  73730
['prune_low_magnitude_block_10_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_10_p  (None, 14, 14, 96)  385
['prune_low_magnitude_block_10_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_11_e  (None, 14, 14, 576)  110594
['prune_low_magnitude_block_10_pr
 xpand (PruneLowMagnitude)
oject_BN[0][0]']

 prune_low_magnitude_block_11_e  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_11_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_11_e  (None, 14, 14, 576)  1
['prune_low_magnitude_block_11_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_11_d  (None, 14, 14, 576)  5185
['prune_low_magnitude_block_11_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_11_d  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_11_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_11_d  (None, 14, 14, 576)  1
['prune_low_magnitude_block_11_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_11_p  (None, 14, 14, 96)  110594
['prune_low_magnitude_block_11_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']
```

```
 prune_low_magnitude_block_11_p  (None, 14, 14, 96)  385
['prune_low_magnitude_block_11_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_11_a  (None, 14, 14, 96)  1
['prune_low_magnitude_block_10_pr
 dd (PruneLowMagnitude)
oject_BN[0][0]',
'prune_low_magnitude_block_11_pr
oject_BN[0][0]']

 prune_low_magnitude_block_12_e  (None, 14, 14, 576)  110594
['prune_low_magnitude_block_11_ad
 xpand (PruneLowMagnitude)                                       d[0][0]']

 prune_low_magnitude_block_12_e  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_12_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_12_e  (None, 14, 14, 576)  1
['prune_low_magnitude_block_12_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_12_d  (None, 14, 14, 576)  5185
['prune_low_magnitude_block_12_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_12_d  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_12_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_12_d  (None, 14, 14, 576)  1
['prune_low_magnitude_block_12_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_12_p  (None, 14, 14, 96)  110594
['prune_low_magnitude_block_12_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_12_p  (None, 14, 14, 96)  385
['prune_low_magnitude_block_12_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']
 prune_low_magnitude_block_12_a  (None, 14, 14, 96)  1
['prune_low_magnitude_block_11_ad
 dd (PruneLowMagnitude)                                          d[0][0]',
'prune_low_magnitude_block_12_pr
oject_BN[0][0]']

 prune_low_magnitude_block_13_e  (None, 14, 14, 576)  110594
['prune_low_magnitude_block_12_ad
 xpand (PruneLowMagnitude)                                       d[0][0]']

 prune_low_magnitude_block_13_e  (None, 14, 14, 576)  2305
['prune_low_magnitude_block_13_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_13_e  (None, 14, 14, 576)  1
['prune_low_magnitude_block_13_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_13_p  (None, 15, 15, 576)  1
['prune_low_magnitude_block_13_ex
 ad (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_13_d  (None, 7, 7, 576)   5185
['prune_low_magnitude_block_13_pa
 epthwise (PruneLowMagnitude)                                    d[0][0]']

 prune_low_magnitude_block_13_d  (None, 7, 7, 576)   2305
['prune_low_magnitude_block_13_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_13_d  (None, 7, 7, 576)   1
['prune_low_magnitude_block_13_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_13_p  (None, 7, 7, 160)   184322
['prune_low_magnitude_block_13_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_13_p  (None, 7, 7, 160)   641
['prune_low_magnitude_block_13_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_14_e  (None, 7, 7, 960)   307202
```

```
['prune_low_magnitude_block_13_pr
 xpand (PruneLowMagnitude)
oject_BN[0][0]']

 prune_low_magnitude_block_14_e  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_14_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_14_e  (None, 7, 7, 960)   1
['prune_low_magnitude_block_14_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_14_d  (None, 7, 7, 960)   8641
['prune_low_magnitude_block_14_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_14_d  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_14_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_14_d  (None, 7, 7, 960)   1
['prune_low_magnitude_block_14_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_14_p  (None, 7, 7, 160)   307202
['prune_low_magnitude_block_14_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_14_p  (None, 7, 7, 160)   641
['prune_low_magnitude_block_14_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_14_a  (None, 7, 7, 160)   1
['prune_low_magnitude_block_13_pr
 dd (PruneLowMagnitude)
oject_BN[0][0]',
'prune_low_magnitude_block_14_pr
oject_BN[0][0]']

 prune_low_magnitude_block_15_e  (None, 7, 7, 960)   307202
['prune_low_magnitude_block_14_ad
 xpand (PruneLowMagnitude)                                       d[0][0]']
 prune_low_magnitude_block_15_e  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_15_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_15_e  (None, 7, 7, 960)   1
['prune_low_magnitude_block_15_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_15_d  (None, 7, 7, 960)   8641
['prune_low_magnitude_block_15_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_15_d  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_15_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_15_d  (None, 7, 7, 960)   1
['prune_low_magnitude_block_15_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_15_p  (None, 7, 7, 160)   307202
['prune_low_magnitude_block_15_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_15_p  (None, 7, 7, 160)   641
['prune_low_magnitude_block_15_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_block_15_a  (None, 7, 7, 160)   1
['prune_low_magnitude_block_14_ad
 dd (PruneLowMagnitude)                                          d[0][0]',
'prune_low_magnitude_block_15_pr
oject_BN[0][0]']

 prune_low_magnitude_block_16_e  (None, 7, 7, 960)   307202
['prune_low_magnitude_block_15_ad
 xpand (PruneLowMagnitude)                                       d[0][0]']

 prune_low_magnitude_block_16_e  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_16_ex
 xpand_BN (PruneLowMagnitude)                                    pand[0][0]']

 prune_low_magnitude_block_16_e  (None, 7, 7, 960)   1
```

```
['prune_low_magnitude_block_16_ex
 xpand_relu (PruneLowMagnitude)                                  pand_BN[0][0]']

 prune_low_magnitude_block_16_d  (None, 7, 7, 960)   8641
['prune_low_magnitude_block_16_ex
 epthwise (PruneLowMagnitude)
pand_relu[0][0]']

 prune_low_magnitude_block_16_d  (None, 7, 7, 960)   3841
['prune_low_magnitude_block_16_de
 epthwise_BN (PruneLowMagnitude                                  pthwise[0][0]']
 )

 prune_low_magnitude_block_16_d  (None, 7, 7, 960)   1
['prune_low_magnitude_block_16_de
 epthwise_relu (PruneLowMagnitu
pthwise_BN[0][0]']
 de)

 prune_low_magnitude_block_16_p  (None, 7, 7, 320)   614402
['prune_low_magnitude_block_16_de
 roject (PruneLowMagnitude)
pthwise_relu[0][0]']

 prune_low_magnitude_block_16_p  (None, 7, 7, 320)   1281
['prune_low_magnitude_block_16_pr
 roject_BN (PruneLowMagnitude)                                   oject[0][0]']

 prune_low_magnitude_Conv_1 (Pr  (None, 7, 7, 1280)  819202
['prune_low_magnitude_block_16_pr
 uneLowMagnitude)
oject_BN[0][0]']

 prune_low_magnitude_Conv_1_bn   (None, 7, 7, 1280)  5121
['prune_low_magnitude_Conv_1[0][0
 (PruneLowMagnitude)                                             ]']

 prune_low_magnitude_out_relu (  (None, 7, 7, 1280)  1
['prune_low_magnitude_Conv_1_bn[0
 PruneLowMagnitude)                                              ][0]']

 prune_low_magnitude_global_ave  (None, 1280)        1
['prune_low_magnitude_out_relu[0]
 rage_pooling2d_2 (PruneLowMagn                                  [0]']
 itude)

 prune_low_magnitude_prediction  (None, 12)          30734
['prune_low_magnitude_global_aver
```

```
 s (PruneLowMagnitude)
age_pooling2d_2[0][0]']

================================================================================
==================
Total params: 4,414,443
Trainable params: 2,239,244
Non-trainable params: 2,175,199
________________________________________________________________________________
__________________
```

One Shot Sparsity appears to have failed

```
[ ]: print_model_weights_sparsity(pruned_model_one_shot)
```

```
Conv1/kernel:0: 0.00% sparsity  (0/864)
expanded_conv_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/288)
expanded_conv_project/kernel:0: 0.00% sparsity  (0/512)
block_1_expand/kernel:0: 0.00% sparsity  (0/1536)
block_1_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/864)
block_1_project/kernel:0: 0.00% sparsity  (0/2304)
block_2_expand/kernel:0: 0.00% sparsity  (0/3456)
block_2_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1296)
block_2_project/kernel:0: 0.00% sparsity  (0/3456)
block_3_expand/kernel:0: 0.00% sparsity  (0/3456)
block_3_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1296)
block_3_project/kernel:0: 0.00% sparsity  (0/4608)
block_4_expand/kernel:0: 0.00% sparsity  (0/6144)
block_4_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1728)
block_4_project/kernel:0: 0.00% sparsity  (0/6144)
block_5_expand/kernel:0: 0.00% sparsity  (0/6144)
block_5_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1728)
block_5_project/kernel:0: 0.00% sparsity  (0/6144)
block_6_expand/kernel:0: 0.00% sparsity  (0/6144)
block_6_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/1728)
block_6_project/kernel:0: 0.00% sparsity  (0/12288)
block_7_expand/kernel:0: 0.00% sparsity  (0/24576)
block_7_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_7_project/kernel:0: 0.00% sparsity  (0/24576)
block_8_expand/kernel:0: 0.00% sparsity  (0/24576)
block_8_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_8_project/kernel:0: 0.00% sparsity  (0/24576)
block_9_expand/kernel:0: 0.00% sparsity  (0/24576)
block_9_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_9_project/kernel:0: 0.00% sparsity  (0/24576)
block_10_expand/kernel:0: 0.00% sparsity  (0/24576)
block_10_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/3456)
block_10_project/kernel:0: 0.00% sparsity  (0/36864)
```

```
block_11_expand/kernel:0: 0.00% sparsity  (0/55296)
block_11_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/5184)
block_11_project/kernel:0: 0.00% sparsity  (0/55296)
block_12_expand/kernel:0: 0.00% sparsity  (0/55296)
block_12_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/5184)
block_12_project/kernel:0: 0.00% sparsity  (0/55296)
block_13_expand/kernel:0: 0.00% sparsity  (0/55296)
block_13_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/5184)
block_13_project/kernel:0: 0.00% sparsity  (0/92160)
block_14_expand/kernel:0: 0.00% sparsity  (0/153600)
block_14_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/8640)
block_14_project/kernel:0: 0.00% sparsity  (0/153600)
block_15_expand/kernel:0: 0.00% sparsity  (0/153600)
block_15_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/8640)
block_15_project/kernel:0: 0.00% sparsity  (0/153600)
block_16_expand/kernel:0: 0.00% sparsity  (0/153600)
block_16_depthwise/depthwise_kernel:0: 0.00% sparsity  (0/8640)
block_16_project/kernel:0: 0.00% sparsity  (0/307200)
Conv_1/kernel:0: 0.00% sparsity  (0/409600)
predictions/kernel:0: 0.00% sparsity  (0/15360)
```

1.15 Discussion of Implementation and Results

To summarize, there was great difficulty in developing an initial model with good validation accuracy. All the combinations of models and alterations include: * AlexNet * Full VMMRdb data set with ~9000 classes * Top 1 Categorical Accuracy * Top 22 Categorical Accuracy * Top 200 Categorical Accuracy * Combining Years into Make/Model Classes With Minimum 100 Images Per Class(363 Classes Total) * Top 1 Categorical Accuracy * Top 10 Categorical Accuracy * MobileNetV2 & MobileNetV3 (Large) * Reduced data set that included classes only with >=400 images (25 classes total) * Accuracy only (top k categorical accuracy wasn't used because of the significant reduction in number of classes) * Hyperparameter tuning, data augmentation, regularization (many combinations attempted), including: * Rescaling, rotation, zoom, width/height shifting, shearing, flipping. * Pretrained weights from ImageNet (unfrozen) * L2 Regularization * Hyperparameter adjustments attempted: * Max/Avergage/None pooling * Alpha between 1 and 3 * Dropout rate between 0.5 and 1 * Adding more ReLu/Convolutional 2D layers * Epochs between 30 and 100 * Initial learning rate between 1e-1 and 1e-10 * Combining 'Year' to Make/Model classes (12 in total) * Increased number of epochs between 60-180 (through re-running the same code set to 60 epochs).

Through all of these considerations, the maximum validation accuracy that could be achieved was approximately 50%.

1.16 Challenges

There were several challenges with this project, mainly with what was assumed to be the 'easy' part (loading a data set into Google Colab, and getting a decent accuracy on a base model).

The size of the data set proved to be too much for the free tier of Google Colab, and required Google Plus compute credits so the full data set could be loaded into the RAM. Prior to upgrading, the data set was uploaded to a Google Drive account and attempted to be read from there, however this resulted in extremely long read times (again due to the size of the data set, but also due to the slow read/write speeds of Google Colab from/to Google Drive).

Once the VMMRdb was able to be loaded into the Google Colab runtime (with a Google Colab Plus account), the next problem became training the AlexNet model. It took an extremely long time to train and heavily used compute credits.

Though the data set had many images (nearly 300,000), There were also many classes (8,174), which made the average number of images per class very low (approx. 34 images per class). While training the AlexNet model was attempted with the full set of images and classes, it should be noted that most papers that used this data set ended up using only the classes that had a large number of images [2][3]. For this reason and the other problems encountered (above), we decided to pivot into using a subsection of the data set and also use a smaller model like the MobileNets.

This pivot proved to be better; training accuracy was able to reach greater than 90% on most occasions (provided training lasted for enough epochs; this was usually in the 100-200 epoch range, but training was performed in groups of 60 epochs, then recomputed if the model seemed like it hadn't plateued yet). Unfortunately, validation accuracy was difficult to improve. There were many techniques attempted, as listed above in the 'Discussion of Implementation and Results' section, however the greated validation accuracy that could be achieved was just approximately 56%.

After this model was saved, it was attempted to be pruned. Following the code from the lab, iterative pruning was successful, but a known keras issue was encountered when trying to use checkpointing in combination with iterative pruning (to preserve the best performing model). One shot pruning appeared to be unsuccessful, and time ran out to continue troubleshooting.

1.17 Discussion of Results

Overall, the main purpose of this project was to explore the feasibility of using a lightweight CNN model to process dashcam data for vehicle make/model/year identification. After the significant amount of problems encountered in this project, it is clear that this task may be too large of a scope for a small scale (compressed) CNN as originally envisioned. We chose the VMMRdb as the data set, mainly because we felt it to be most representative of the diversity of vehicles a dashcam would be exposed to in the real world. Given the huge diversity of vehicles on the road today, it may be more practical for a dashcam vehicle identifier to focus on a limited number of classes of vehicles.

1.18 Future Work

Future improvements include better sorting of the dataset, colour augmentation and implementation of other algorithms, such as the YOLO Architecture for better detection of vehicles within a "busy" image with other objects present.

With colour augmentation, the intention would be to set images to grayscale, and thus force the model to classify the vehicles based off their size and shape rather than colour. This could rule out the potential suspicion that the colour of the vehicles is somehow contirbuting to the confusion and poor classifications of the model.

In addition, upon further researching into the task itself, the YOLO architecture may be favourable for the variety of images in this task. Originally this architecture was not pursued as most images appeared to be relatively “isolated” and the main focal point of the image. However, with YOLO, perhaps the model could better deal with the variety of backgrounds (streets, garages etc), as well as angles from which the image is taken.

An immediate next step could also include going back to the AlexNet model and using the reduced and filtered dataset to re-train the model. This step was omitted due to lack of time and compute credits available to compile the large AlexNet model.

Lastly, a data set that identifies only a portion of the vehicle from a standard view (e.g. rear view only) may be a more practical approach, rather than a data set that includes vehicles from all angles. This may allow the CNN to make better distinctions between features that belong to each class.

1.19 References

[1] F. Tafazzoli, H. Frigui and K. Nishiyama, “A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition,” in Conference on Computer Vision and Patter Recognition (CVPR), Honolulu, Hawaii, 2017.

[2] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In NIPS, pp. 1106–1114, 2012.

[3] Kristiani, E., Yang, C. T., & Huang, C. Y. (2020). iSEC: an optimized deep learning model for image classification on edge computing. IEEE Access, 8, 27267-27276.