

# VegNet - Image Classification

India's largest fresh produce supply chain company is a pioneer in solving one of the toughest supply chain challenges by leveraging innovative technology. They source fresh produce from farmers and deliver it to businesses within 12 hours. An integral component of their automation process involves developing robust classifiers that can distinguish between images of different types of vegetables, as well as accurately label images that do not contain any specific type of vegetable as noise.

As a starting point, The company has provided us with a dataset scraped from the web which contains train and test folders, each having 4 sub-folders with images of onions, potatoes, tomatoes and some market scenes. We have been tasked with preparing a multiclass classifier for identifying these vegetables. The dataset provided has all the required images to achieve the task.

**Dataset Link:** [https://drive.google.com/file/d/1cIZX-IV\\_MLxKHSyeyTheX5OCQtNCUcqT/view?usp=sharing](https://drive.google.com/file/d/1cIZX-IV_MLxKHSyeyTheX5OCQtNCUcqT/view?usp=sharing)

## Context

This dataset contains images of the following food items: noise-Indian market and images of vegetables-onion, potato and tomato.

## Data Collection

The images in this dataset were scraped from Google.

## Content

This dataset contains a folder train, which has a total of 3135 images, split into four folders as follows:

Tomato : 789

Potato : 898

Onion : 849

Indian market : 599

This dataset contains another folder test which has a total of 351 images, split into four folders

Tomato : 106

potato : 83

onion : 81

Indian market : 81

## Objective:

The objective is to develop a program that can recognize the vegetable item(s) in a photo and identify them for the user.

## Importing necessary libraries

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import tensorflow as tf
import keras
```

### Loading the dataset

```
In [2]: main_directory='./nиняcart_data/nиняcart_data'
train_data,validation_data = tf.keras.utils.image_dataset_from_directory(directory=f'{main_directory}/train')
test_data = tf.keras.utils.image_dataset_from_directory(directory=f'{main_directory}/tes
Found 3135 files belonging to 4 classes.
Using 2508 files for training.
Using 627 files for validation.
Found 351 files belonging to 4 classes.
```

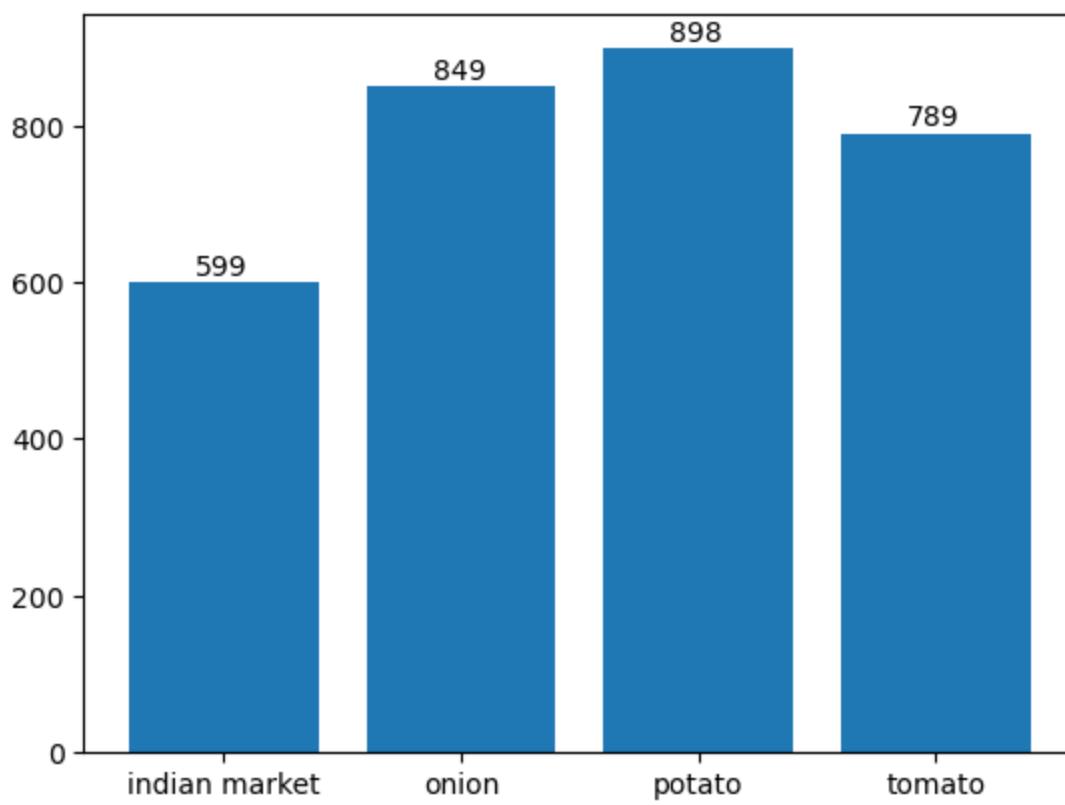
### Unique Classes in the dataset

```
In [3]: train_data.class_names
Out[3]: ['indian market', 'onion', 'potato', 'tomato']
```

The dataset contains different images of markets in India and vegetables like Onion, Potato, and Tomato.

### Checking the number of images under each class

```
In [4]: import os
class_files = {}
for classes in train_data.class_names:
    directory = f'{main_directory}/train/{classes}'
    num_files = len(os.listdir(directory))
    class_files[classes] = num_files
print()
ax = plt.bar(x=class_files.keys(), height=class_files.values())
for patch in ax.patches:
    plt.annotate(patch.get_height(), (patch.get_x() + patch.get_width() * 0.35, patch.get_heig
plt.show()
```



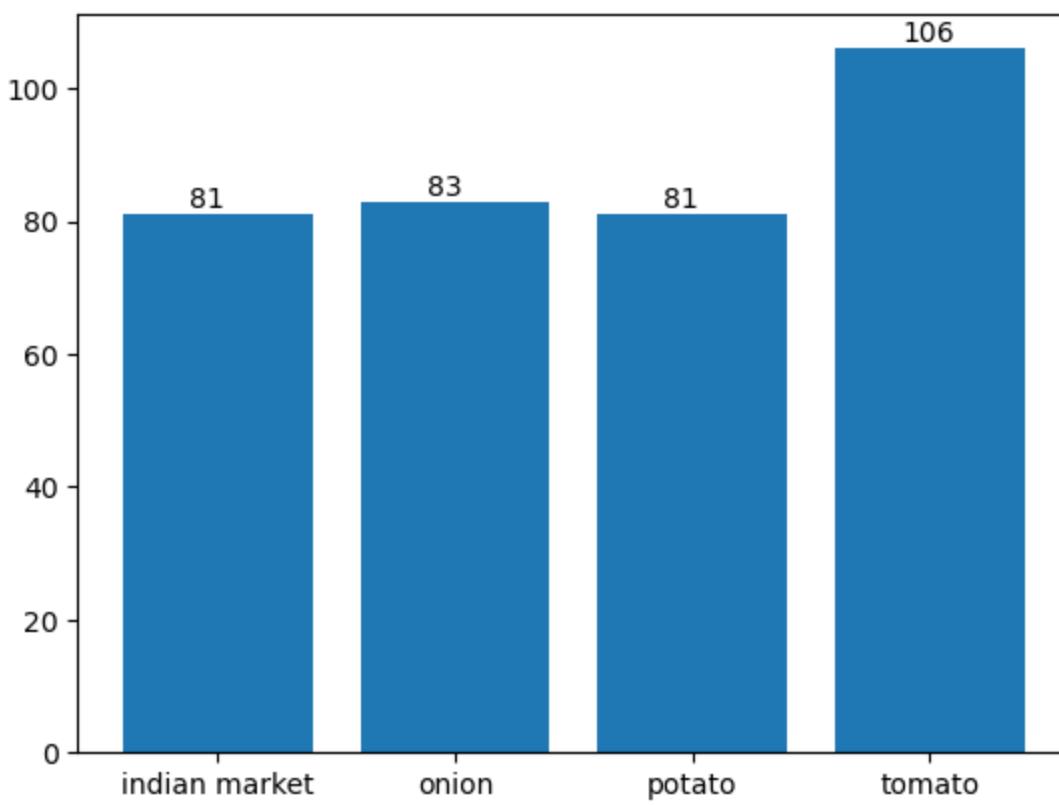
From the above, we can see that there are about 599 images under Indian market class, 849 images under Onion class, 898 images under Potato class and about 789 images under Tomato class.

```
In [5]: class_files
```

```
Out[5]: {'indian market': 599, 'onion': 849, 'potato': 898, 'tomato': 789}
```

These are split into training and validation dataset to train and validate the CNN model.

```
In [6]: test_files = {}
for classes in test_data.class_names:
    directory = f'{main_directory}/test/{classes}'
    num_files = len(os.listdir(directory))
    test_files[classes] = num_files
print()
ax = plt.bar(x=test_files.keys(), height=test_files.values())
for patch in ax.patches:
    plt.annotate(patch.get_height(), (patch.get_x() + patch.get_width() * 0.35, patch.get_heig
plt.show()
```

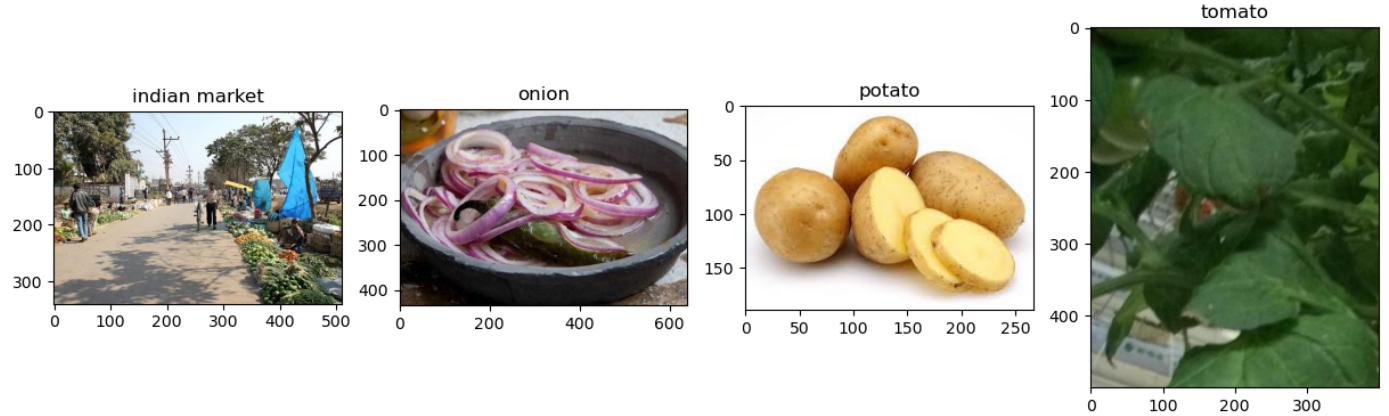


And in the Testing dataset, there are about 81 images under Indian market class, 83 images under Onion class, 81 images under Potato class and about 106 images under Tomato class.

### Displaying few samples of the dataset

In [7]:

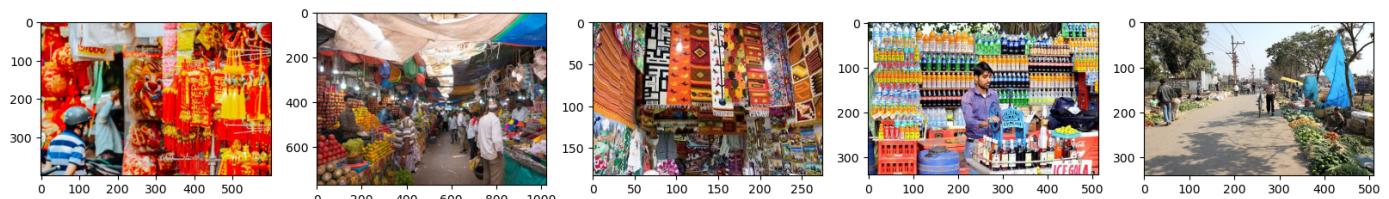
```
import os
num=1
plt.figure(figsize=(15,5))
for classes in train_data.class_names:
    directory = f'{main_directory}/train/{classes}'
    files = os.listdir(directory)
    img=files[-1]
    img = tf.keras.preprocessing.image.load_img(f'{directory}/{img}')
    plt.suptitle('Image samples from the loaded dataset')
    plt.subplot(1,4,num)
    plt.title(f'{classes}')
    plt.imshow(img)
    num+=1
    if num>=5:
        break
plt.show()
```



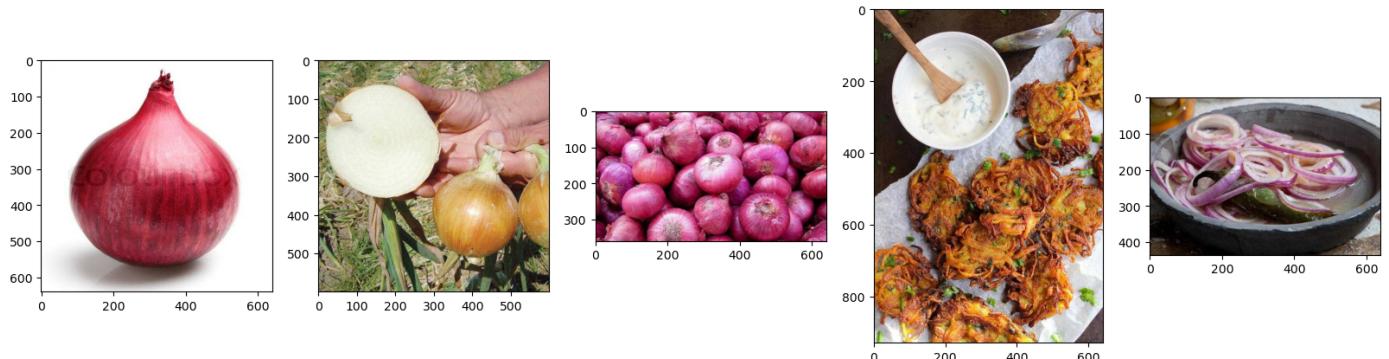
### Display 5 Samples under each classes present in the dataset

```
In [8]: num=1
for classes in train_data.class_names:
    directory = f'{main_directory}/train/{classes}'
    files = os.listdir(directory)
    images=files[-5:]
    images = [tf.keras.preprocessing.image.load_img(f'{directory}/{img}') for img in images]
    plt.figure(figsize=(20,5))
    for i in range(len(images)):
        plt.suptitle(f'5 Samples from {classes} class')
        plt.subplot(1,5,i+1)
        plt.imshow(images[i])
    plt.show()
```

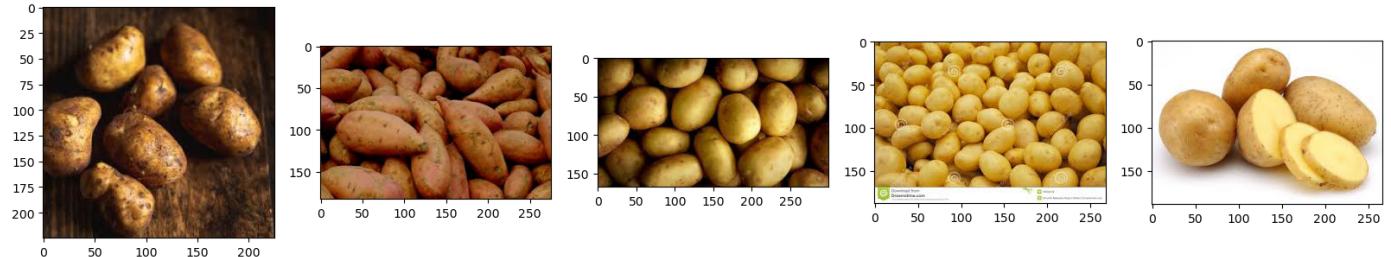
5 Samples from indian market class



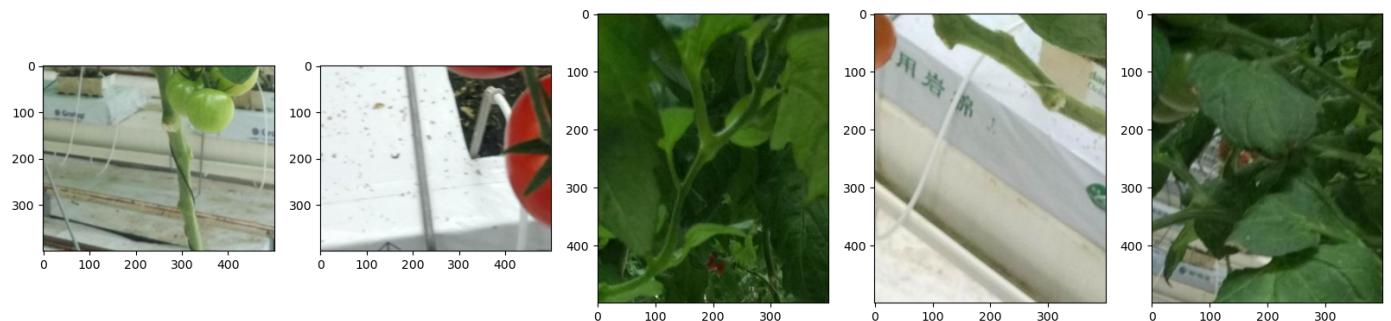
5 Samples from onion class



5 Samples from potato class



5 Samples from tomato class

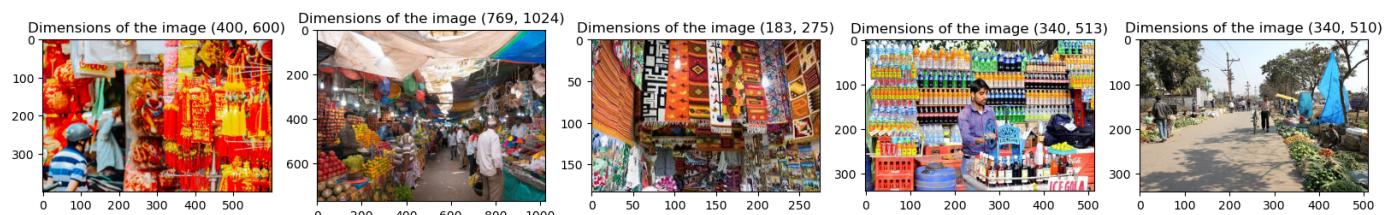


## Plotting images to check their dimensions

```
In [9]: num=1
for classes in train_data.class_names:
    directory = f'{main_directory}/train/{classes}'
    files = os.listdir(directory)
    images=files[-5:]
    images = [tf.keras.preprocessing.image.load_img(f'{directory}/{img}') for img in images]
    plt.figure(figsize=(20,5))
    for i in range(len(images)):
        plt.suptitle(f'5 Samples from {classes} class')
        img_height=images[i].height
        img_width=images[i].width
        plt.subplot(1,5,i+1)
        plt.title(f'Dimensions of the image {(img_height,img_width)}')
        plt.imshow(images[i])

    plt.show()
```

5 Samples from indian market class





From the above plot, we can see that the dimensions of the images are not same. The dimensions has to be same inorder to train the CNN model

### Defining the Preprocessing steps for the images.

Lets keep the dimensions of the image restricted to 128 X 128. ie: Height = 128 and Width = 128. Then also perform rescale operation to restrict the pixel values between 0 and 1.

```
In [10]: from tensorflow.keras.layers import Resizing, Rescaling
from tensorflow.keras.models import Sequential
```

```
In [11]: def preprocess_image(train_data, validation_data, test_data, height=128, width=128):
    img_preprocess = Sequential(name='preprocess_image')
    img_preprocess.add(Resizing(height=height, width=width))
    img_preprocess.add(Rescaling(1.0/255))

    train_ds = train_data.map(lambda x,y : (img_preprocess(x),y), num_parallel_calls=tf.data.experimental.AUTOTUNE)
    validation_ds = validation_data.map(lambda x,y:(img_preprocess(x),y), num_parallel_calls=tf.data.experimental.AUTOTUNE)
    test_ds = test_data.map(lambda x,y:(img_preprocess(x),y), num_parallel_calls=tf.data.experimental.AUTOTUNE)

    return train_ds, validation_ds, test_ds
```

```
In [12]: train_ds, validation_ds, test_ds = preprocess_image(train_data, validation_data, test_data)
```

## Lets store each of the model architectures in a dictionary

Lets store each of built model in a dictionary so that it would be easier for us to compare the results during inference.

```
In [13]: built_models = {}
```

### Building the CNN Model

Lets build a Simple CNN model using the keras sequential API with convolution layers to do the image classification.

```
In [14]: def model_arch1(height=128,width=128,channels=3):
    model = Sequential(
        name='cnn_model1',
        layers = [
            tf.keras.layers.Conv2D(filters=16,kernel_size=3,activation='relu',use_bias=True),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=32,kernel_size=3,activation='relu',use_bias=True),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=64,kernel_size=3,activation='relu',use_bias=True),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=128,kernel_size=3,activation='relu',use_bias=True),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=256,kernel_size=3,activation='relu',use_bias=True),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.GlobalAveragePooling2D(), #Convolutional layers end here.
            tf.keras.layers.Dense(units=128,activation='relu',use_bias=True),
            tf.keras.layers.Dense(units=64,activation='relu',use_bias=True),
            tf.keras.layers.Dense(units=4,activation='softmax',use_bias=True)
        ]
    )
    return model
```

```
In [15]: model = model_arch1()
model.summary()
```

```
C:\Anaconda\lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning:
  ng: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential
  models, prefer using an `Input(shape)` object as the first layer in the model instead.
    super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

**Model: "cnn\_model1"**

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 16)	448
max_pooling2d (MaxPooling2D)	(None, 63, 63, 16)	0
conv2d_1 (Conv2D)	(None, 61, 61, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	73,856
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 128)	0

conv2d_4 (Conv2D)	(None, 4, 4, 256)	295,168
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 256)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 256)	0
dense (Dense)	(None, 128)	32,896
dense_1 (Dense)	(None, 64)	8,256
dense_2 (Dense)	(None, 4)	260

Total params: 434,020 (1.66 MB)

Trainable params: 434,020 (1.66 MB)

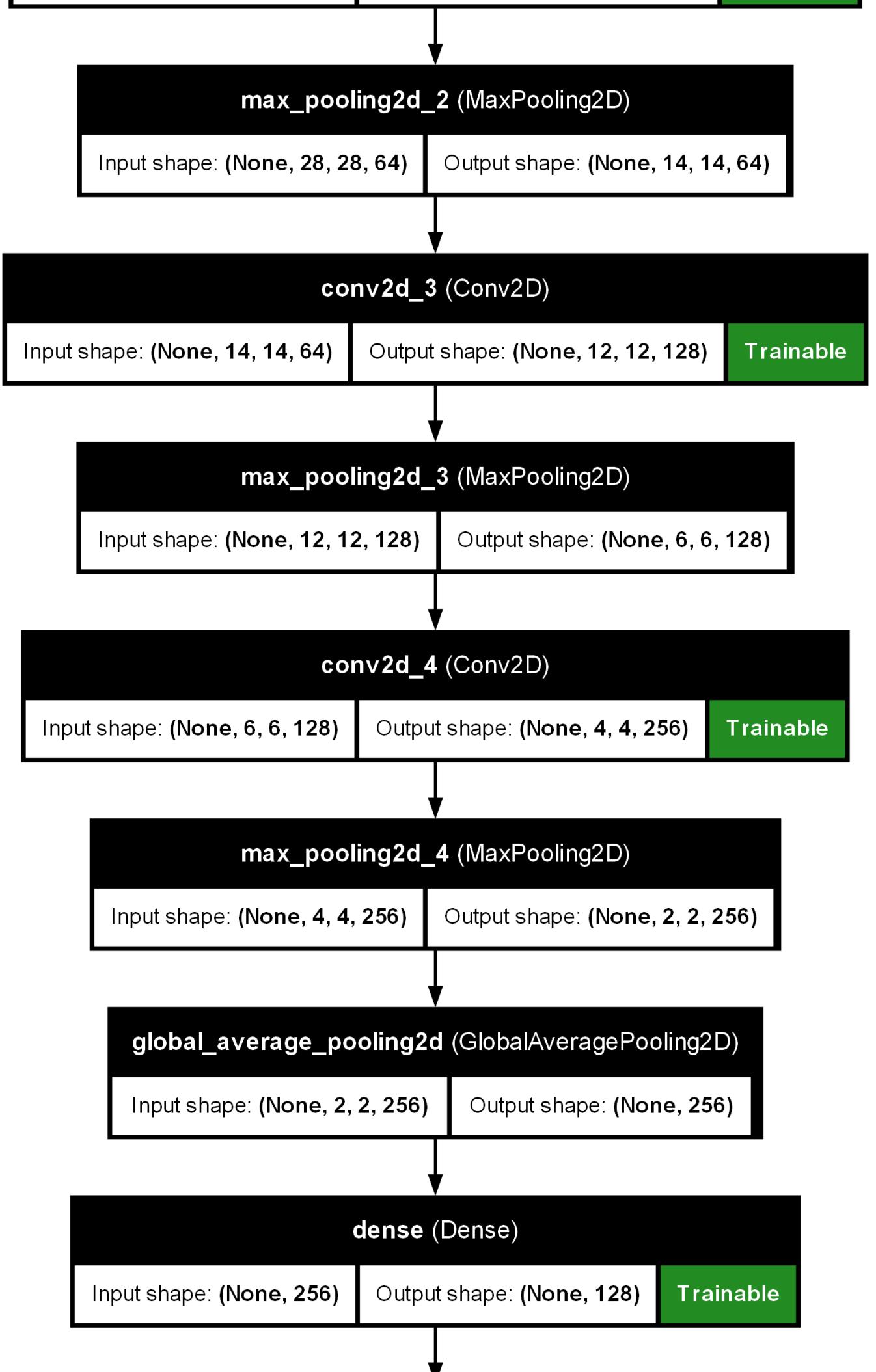
Non-trainable params: 0 (0.00 B)

Visualizing the model

```
In [16]: tf.keras.utils.plot_model(model, show_shapes=True, show_layer_names=True, show_trainable=True)
```

Out[16]:











```
79/79 0s 162ms/step - accuracy: 0.9673 - loss: 0.0994
Epoch 32: val_accuracy did not improve from 0.89633
79/79 15s 190ms/step - accuracy: 0.9671 - loss: 0.099
7 - val_accuracy: 0.8565 - val_loss: 0.4654
```

## Load the best model

```
In [20]: model.load_weights(filepath=f'{direc}/model1_chkpt/best_model.weights.h5')
```

## Checking and Plotting Training and Validation accuracy

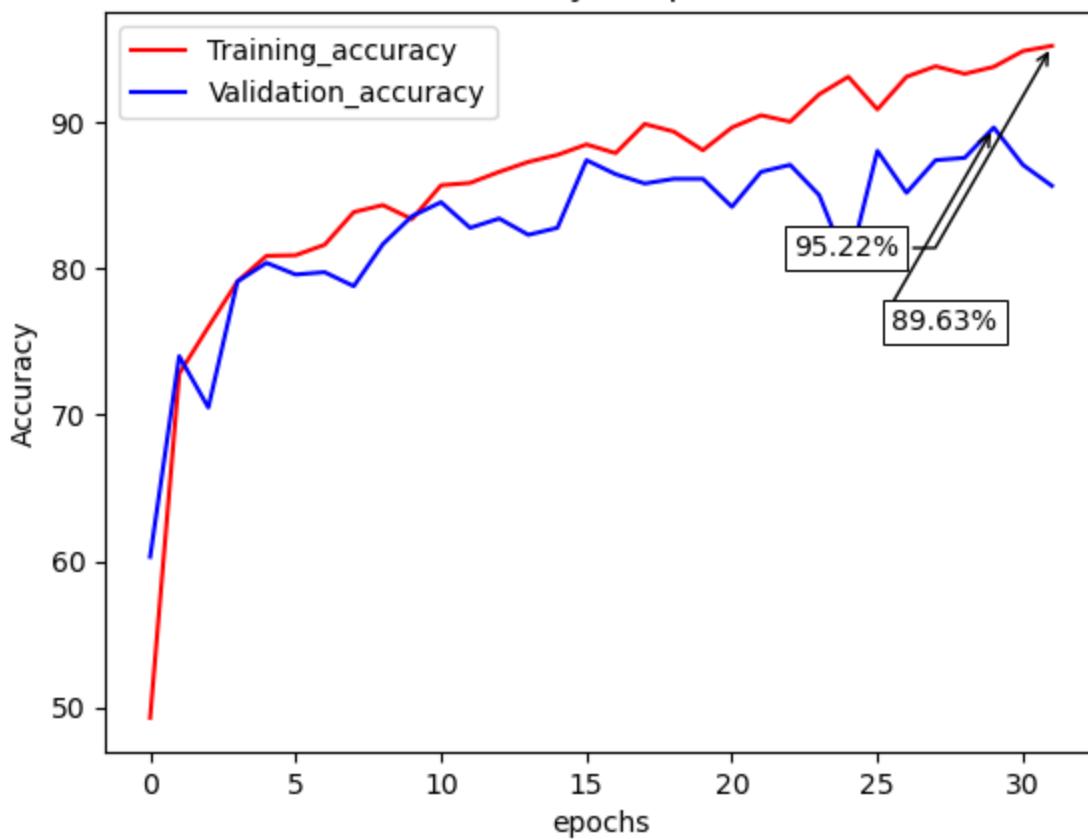
```
In [21]: def annotate_best_accuracy(x,y,xytext,ax=None):
    xmax=x[np.argmax(y)]
    ymax = max(y)
    text = f'{ymax:.2f}%'
    if ax is None:
        ax = plt.gca()
    bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
    arrowprops=dict(arrowstyle="->",connectionstyle="angle,angleA=0,angleB=60")
    kw = dict(xycoords='data',textcoords="axes fraction",
              arrowprops=arrowprops, bbox=bbox_props, ha="right", va="top")
    ax.annotate(text, xy=(xmax, ymax), xytext=xytext, **kw)
```

```
In [22]: def plot_train_val_accuracy(fitted_model):
    epochs = range(0,len(fitted_model.history['accuracy']))
    training_accuracy = [acc*100 for acc in fitted_model.history['accuracy']]
    validation_accuracy = [acc*100 for acc in fitted_model.history['val_accuracy']]

    plt.plot(epochs,training_accuracy,label='Training_accuracy',color='red')
    annotate_best_accuracy(epochs,training_accuracy,xytext=(0.8,0.7))
    plt.plot(epochs,validation_accuracy,label='Validation_accuracy',color='blue')
    annotate_best_accuracy(epochs,validation_accuracy,xytext=(0.9,0.6))
    plt.legend()
    plt.title('Accuracy vs Epochs')
    plt.xlabel('epochs')
    plt.ylabel('Accuracy')
    plt.show()
```

```
In [23]: plot_train_val_accuracy(fitted_model)
```

## Accuracy vs Epochs



### Prediction on the Test dataset.

```
In [24]: from sklearn.metrics import accuracy_score, f1_score, confusion_matrix, classification_r
```

```
In [25]: def print_accuracy(model,data,show=True):
    true_vals = tf.concat([y for x,y in data],axis=0)
    true_categs = tf.argmax(true_vals, axis=1)
    predicted_vals = model.predict(data,verbose=0)
    predicted_categs = tf.argmax(predicted_vals, axis=1)
    if show:
        print(f'The accuracy of the model on the provided dataset is {accuracy_score(true_vals,predicted_vals)*100} %')
    return (accuracy_score(true_categs,predicted_categs)*100)
```

```
In [26]: print_accuracy(model,test_ds)
```

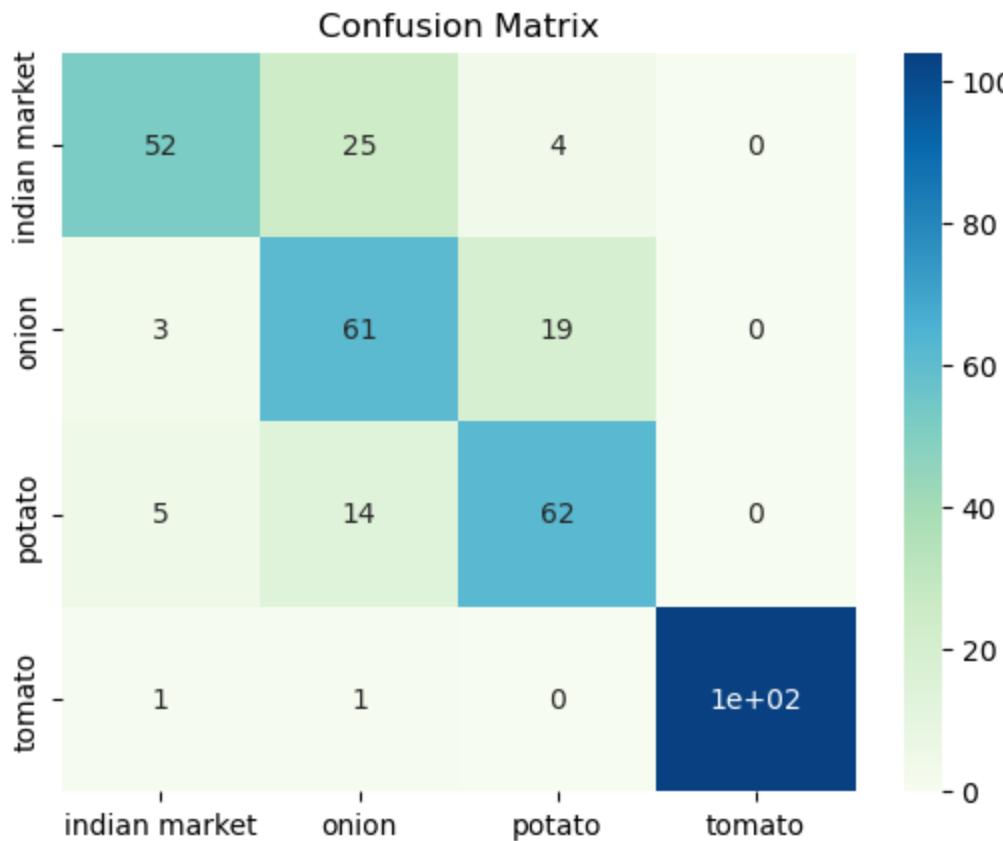
The accuracy of the model on the provided dataset is 79.49%

```
Out[26]: 79.48717948717949
```

### Confusion matrix

```
In [27]: def print_confusion_matrix(model,data,show_plot=False):
    true_vals = tf.concat([y for x,y in data],axis=0)
    true_categs = tf.argmax(true_vals, axis=1)
    predictions = model.predict(data,verbose=0)
    predicted_categs = tf.argmax(predictions, axis=1)
    classNames = test_data.class_names
    conf_mat = confusion_matrix(true_categs,predicted_categs)
    if show_plot:
        plt.title('Confusion Matrix')
        conf_heatmap = sns.heatmap(conf_mat, annot=True, xticklabels=classNames, yticklabels=classNames)
        plt.show()
    else:
        return conf_mat
```

```
In [28]: print_confusion_matrix(model,test_ds,show_plot=True)
```



From the above plots we can clearly see that the model got overfitted with the Training data. We can fine tune this model by including some regularization techniques like kernel regularizers, Batch normalization and Image augmentation. This makes the model more robust to the incoming data.

```
In [29]: built_models['basic_model'] = model
```

## Incorporating Regularisation Techniques in CNN Model

Implementing Kernel regularisation and Batch Normalisation

```
In [30]: from tensorflow.keras.regularizers import L2
def model_arch2(height=128,width=128,channels=3):
    model = Sequential(
        name='cnn_model2',
        layers =
            tf.keras.layers.Conv2D(filters=16,kernel_size=3,use_bias=True,input_shape=(h
            tf.keras.layers.BatchNormalization(),
            tf.keras.layers.Activation(activation='relu'),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=32,kernel_size=3,use_bias=True,kernel_regularizer=L2,
            tf.keras.layers.BatchNormalization(),
            tf.keras.layers.Activation(activation='relu'),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=64,kernel_size=3,use_bias=True,kernel_regularizer=L2,
            tf.keras.layers.BatchNormalization(),
            tf.keras.layers.Activation(activation='relu'),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=128,kernel_size=3,use_bias=True,kernel_regularizer=L2,
            tf.keras.layers.BatchNormalization(),
            tf.keras.layers.Activation(activation='relu'),
            tf.keras.layers.MaxPooling2D(),
            tf.keras.layers.Conv2D(filters=256,kernel_size=3,use_bias=True,kernel_regularizer=L2,
            tf.keras.layers.BatchNormalization(),
```

```
        tf.keras.layers.Activation(activation='relu'),
        tf.keras.layers.MaxPooling2D(),
        tf.keras.layers.GlobalAveragePooling2D(),    #Convolutional layers end here.
        tf.keras.layers.Dense(units=128,use_bias=True,kernel_regularizer=L2(1e-3)),
        tf.keras.layers.BatchNormalization(),
        tf.keras.layers.Activation(activation='relu'),
        tf.keras.layers.Activation(activation='relu'),
        tf.keras.layers.Dense(units=4,activation='softmax',use_bias=True)
    ]
)
return model
```

```
In [31]: model = model_arch2()
model.summary()
```

C:\Anaconda\lib\site-packages\keras\src\layers\convolutional\base\_conv.py:107: UserWarning:  
 ng: Do not pass an `input\_shape` / `input\_dim` argument to a layer. When using Sequential  
 models, prefer using an `Input(shape)` object as the first layer in the model instead.  
 super().\_\_init\_\_(activity\_regularizer=activity\_regularizer, \*\*kwargs)

Model: "cnn\_model12"

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 126, 126, 16)	448
batch_normalization (BatchNormalization)	(None, 126, 126, 16)	64
activation (Activation)	(None, 126, 126, 16)	0
max_pooling2d_5 (MaxPooling2D)	(None, 63, 63, 16)	0
conv2d_6 (Conv2D)	(None, 61, 61, 32)	4,640
batch_normalization_1 (BatchNormalization)	(None, 61, 61, 32)	128
activation_1 (Activation)	(None, 61, 61, 32)	0
max_pooling2d_6 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_7 (Conv2D)	(None, 28, 28, 64)	18,496
batch_normalization_2 (BatchNormalization)	(None, 28, 28, 64)	256
activation_2 (Activation)	(None, 28, 28, 64)	0
max_pooling2d_7 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_8 (Conv2D)	(None, 12, 12, 128)	73,856
batch_normalization_3 (BatchNormalization)	(None, 12, 12, 128)	512
activation_3 (Activation)	(None, 12, 12, 128)	0
max_pooling2d_8 (MaxPooling2D)	(None, 6, 6, 128)	0
conv2d_9 (Conv2D)	(None, 4, 4, 256)	295,168
batch_normalization_4	(None, 4, 4, 256)	1,024

(BatchNormalization)		
activation_4 (Activation)	(None, 4, 4, 256)	0
max_pooling2d_9 (MaxPooling2D)	(None, 2, 2, 256)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32,896
batch_normalization_5 (BatchNormalization)	(None, 128)	512
activation_5 (Activation)	(None, 128)	0
activation_6 (Activation)	(None, 128)	0
dense_4 (Dense)	(None, 4)	516

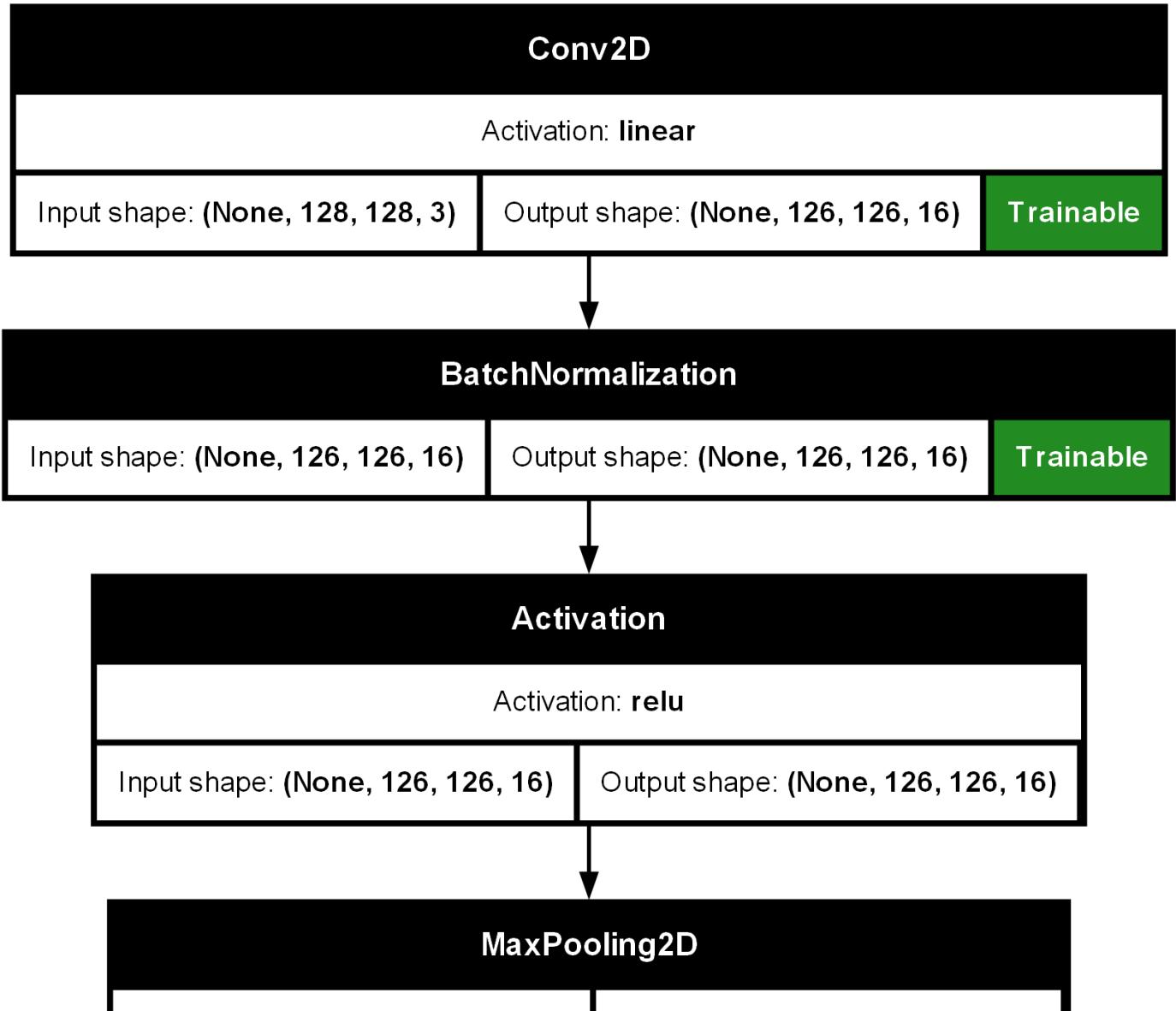
Total params: 428,516 (1.63 MB)

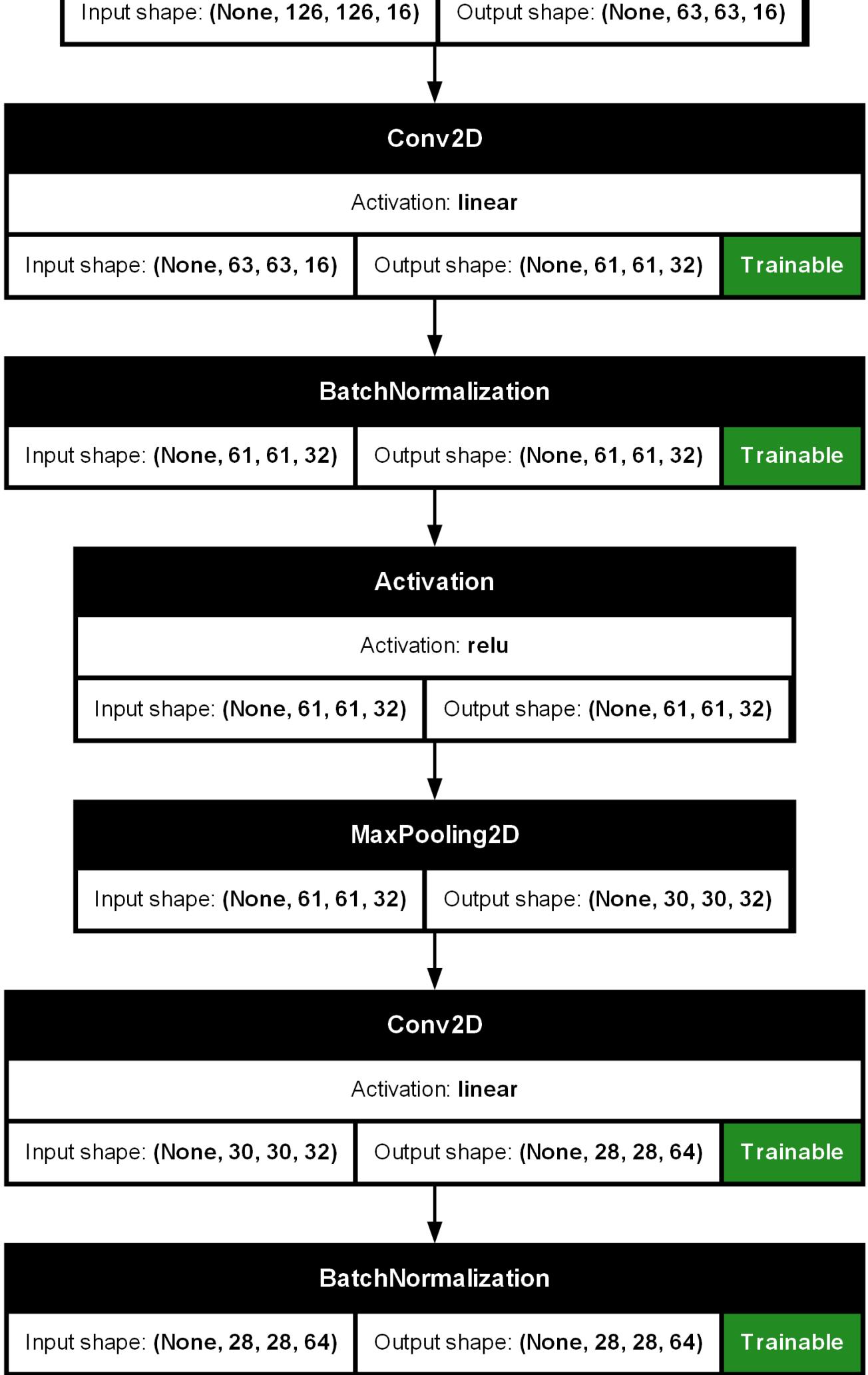
Trainable params: 427,268 (1.63 MB)

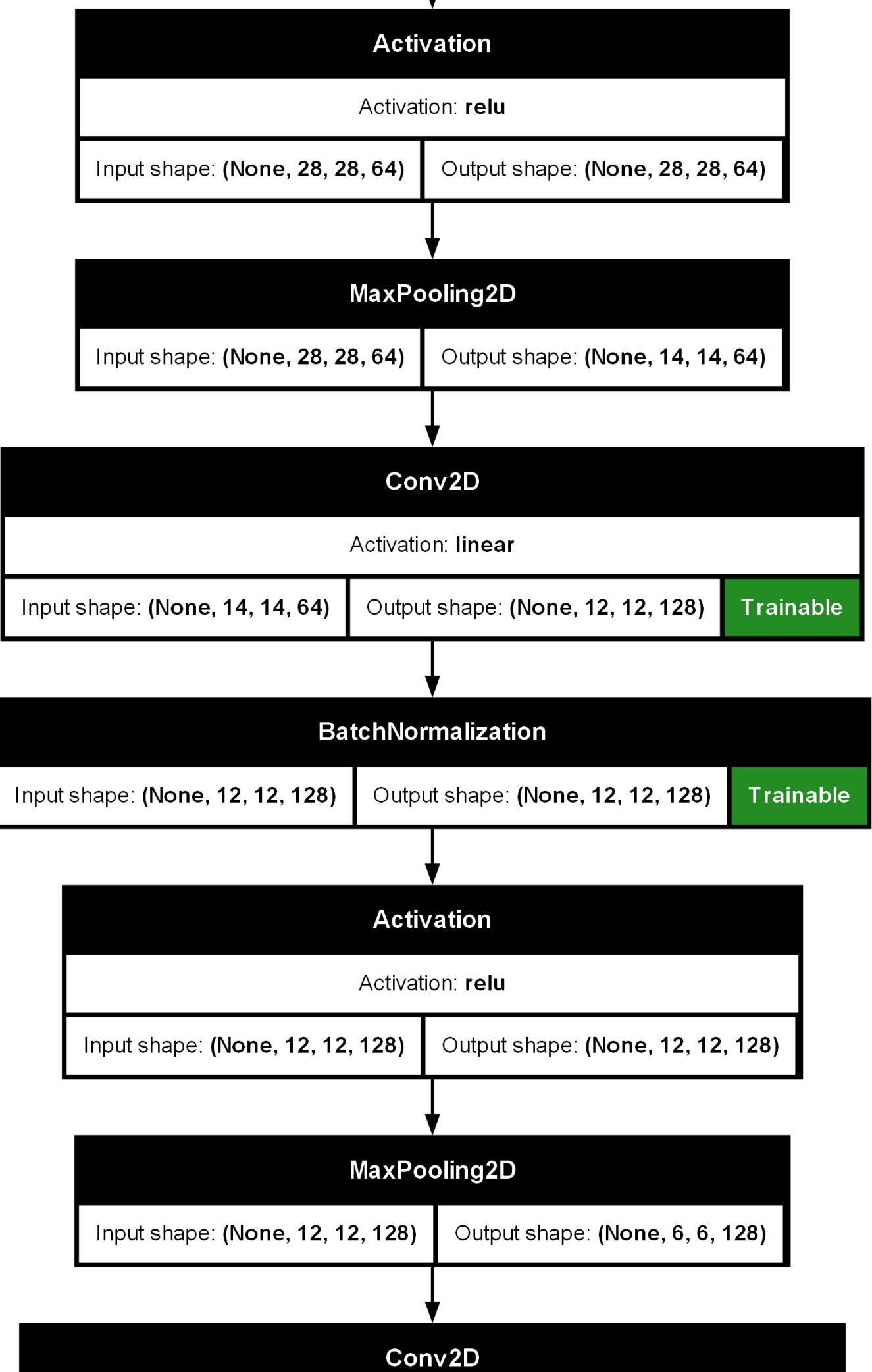
Non-trainable params: 1,248 (4.88 KB)

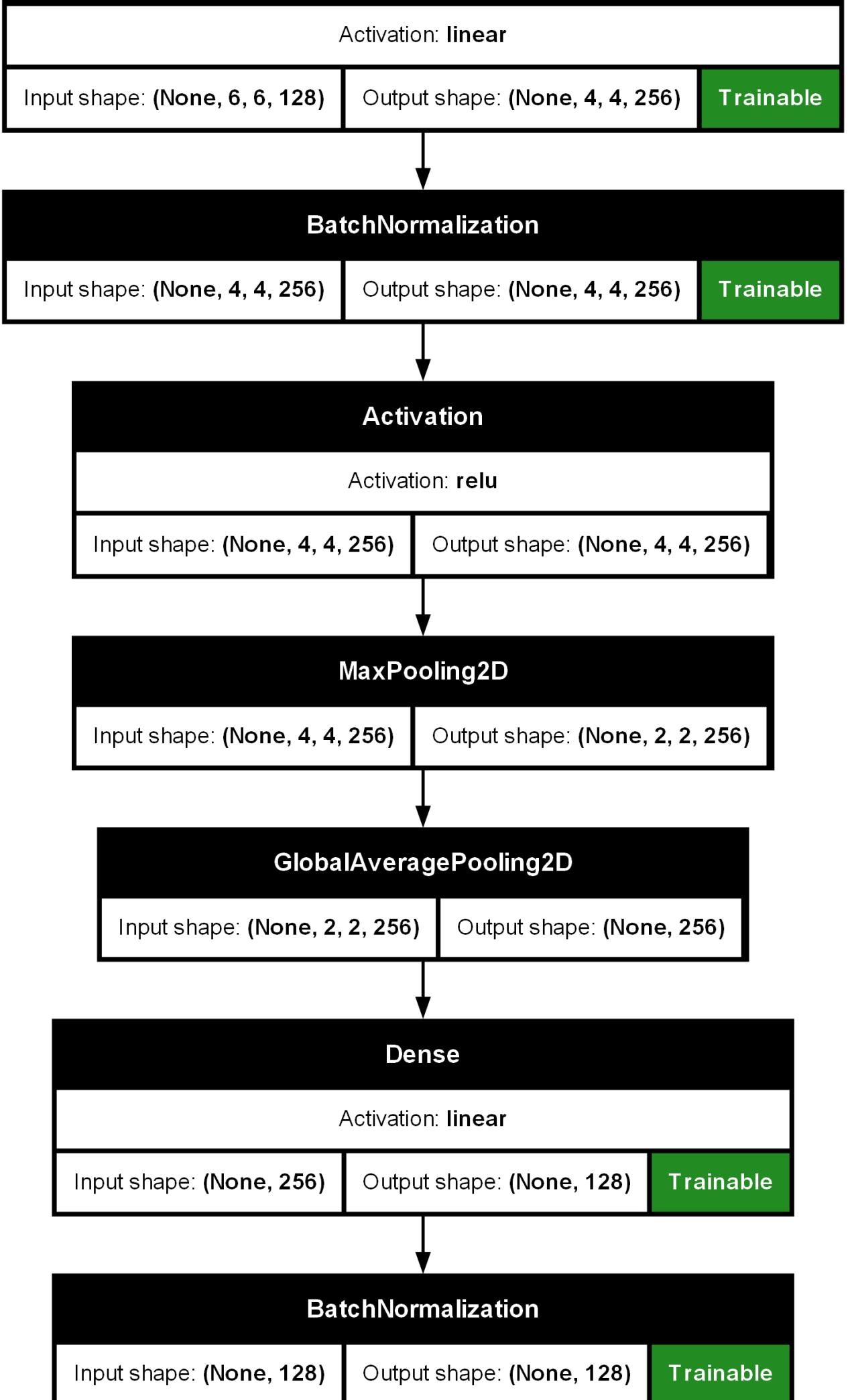
In [32]: `tf.keras.utils.plot_model(model, show_trainable=True, show_shapes=True, show_layer_activati`

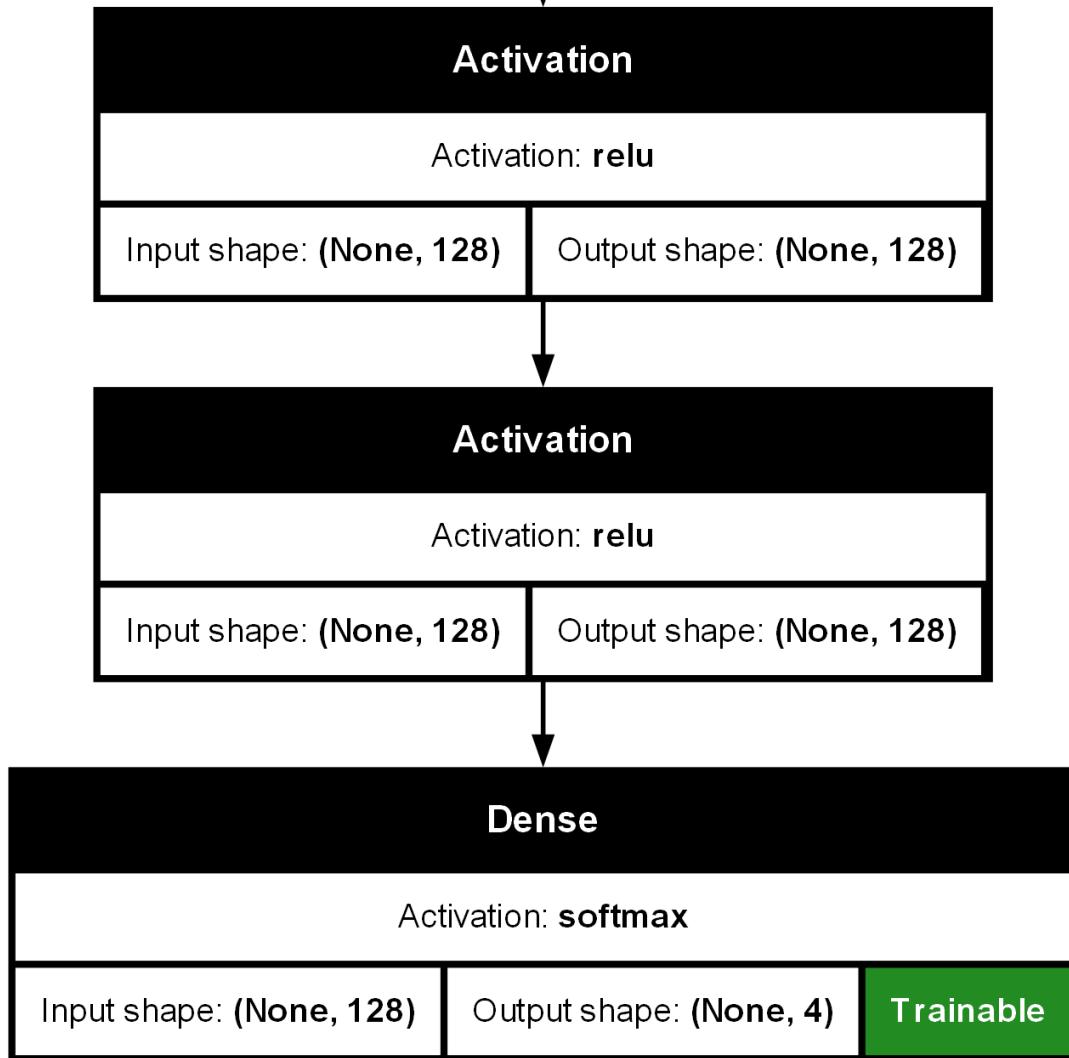
Out[32]:











### Compiling and fitting the model

```

In [33]: model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
direc = './model2_callbacks'
model_chkpt = tf.keras.callbacks.ModelCheckpoint(filepath=f'{direc}/model2_chkpt/best_mo
model_tensorboard = tf.keras.callbacks.TensorBoard(log_dir=f'{direc}/model2_tensorboard'
model_earlyStopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', min_delta=0.00
callbacks=[model_chkpt, model_tensorboard, model_earlyStopping]
regularized_fitted_model = model.fit(train_ds, batch_size=32, epochs=10, validation_data=va

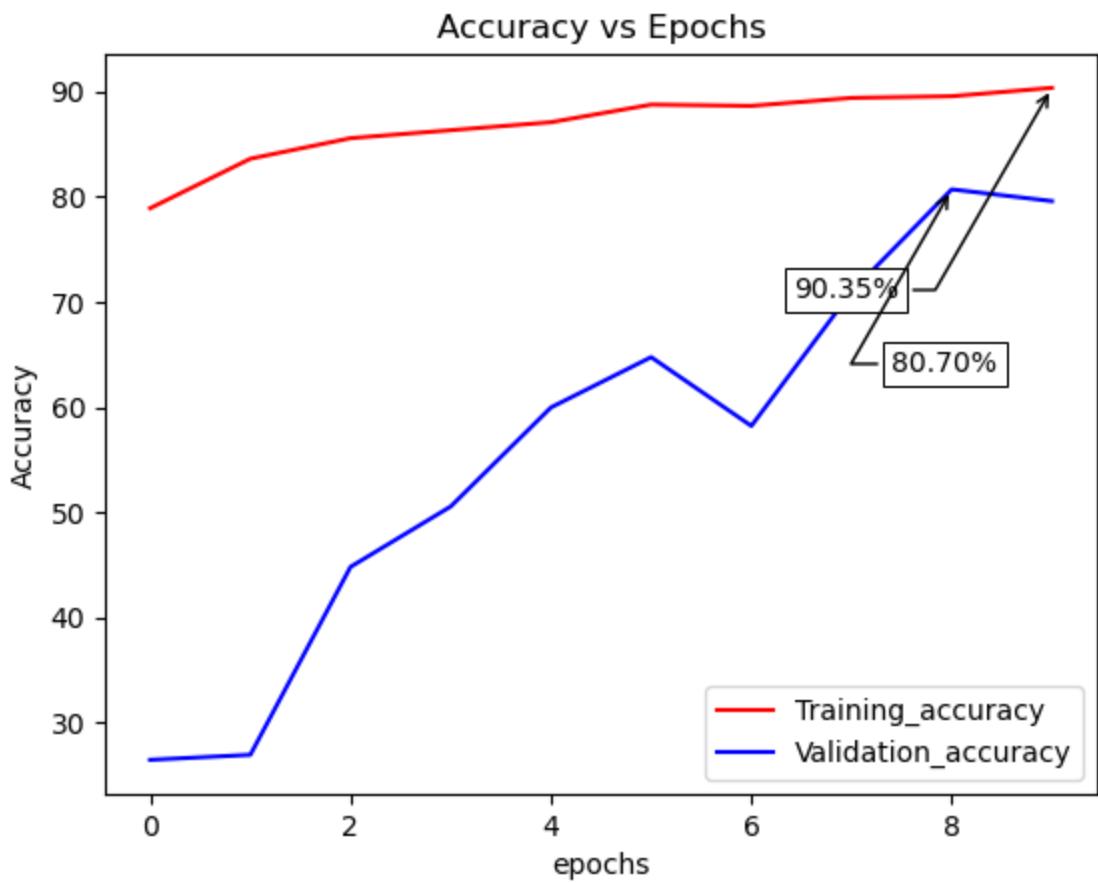
```

Epoch 1/10  
**79/79** 0s 332ms/step - accuracy: 0.7199 - loss: 1.2295  
Epoch 1: val\_accuracy improved from -inf to 0.26475, saving model to ./model2\_callbacks/  
model2\_chkpt/best\_model.weights.h5  
**79/79** 39s 366ms/step - accuracy: 0.7208 - loss: 1.227  
5 - val\_accuracy: 0.2648 - val\_loss: 2.1541  
Epoch 2/10  
**79/79** 0s 315ms/step - accuracy: 0.8366 - loss: 0.9015  
Epoch 2: val\_accuracy improved from 0.26475 to 0.26954, saving model to ./model2\_callba  
ks/model2\_chkpt/best\_model.weights.h5  
**79/79** 27s 343ms/step - accuracy: 0.8366 - loss: 0.901  
5 - val\_accuracy: 0.2695 - val\_loss: 2.8212  
Epoch 3/10  
**79/79** 0s 321ms/step - accuracy: 0.8593 - loss: 0.8245  
Epoch 3: val\_accuracy improved from 0.26954 to 0.44817, saving model to ./model2\_callba  
ks/model2\_chkpt/best\_model.weights.h5  
**79/79** 28s 346ms/step - accuracy: 0.8593 - loss: 0.824  
4 - val\_accuracy: 0.4482 - val\_loss: 2.2770

Epoch 4/10  
**79/79** 0s 315ms/step - accuracy: 0.8692 - loss: 0.7561  
Epoch 4: val\_accuracy improved from 0.44817 to 0.50558, saving model to ./model2\_callbacks/model2\_chkpt/best\_model.weights.h5  
**79/79** 27s 340ms/step - accuracy: 0.8692 - loss: 0.7561  
2 - val\_accuracy: 0.5056 - val\_loss: 1.8406  
Epoch 5/10  
**79/79** 0s 313ms/step - accuracy: 0.8814 - loss: 0.6951  
Epoch 5: val\_accuracy improved from 0.50558 to 0.59968, saving model to ./model2\_callbacks/model2\_chkpt/best\_model.weights.h5  
**79/79** 27s 337ms/step - accuracy: 0.8813 - loss: 0.6951  
4 - val\_accuracy: 0.5997 - val\_loss: 1.4016  
Epoch 6/10  
**79/79** 0s 315ms/step - accuracy: 0.8877 - loss: 0.6421  
Epoch 6: val\_accuracy improved from 0.59968 to 0.64753, saving model to ./model2\_callbacks/model2\_chkpt/best\_model.weights.h5  
**79/79** 27s 340ms/step - accuracy: 0.8877 - loss: 0.6421  
1 - val\_accuracy: 0.6475 - val\_loss: 1.5638  
Epoch 7/10  
**79/79** 0s 315ms/step - accuracy: 0.8953 - loss: 0.6139  
Epoch 7: val\_accuracy did not improve from 0.64753  
**79/79** 27s 339ms/step - accuracy: 0.8952 - loss: 0.6139  
8 - val\_accuracy: 0.5821 - val\_loss: 2.0126  
Epoch 8/10  
**79/79** 0s 318ms/step - accuracy: 0.8914 - loss: 0.5837  
Epoch 8: val\_accuracy improved from 0.64753 to 0.70973, saving model to ./model2\_callbacks/model2\_chkpt/best\_model.weights.h5  
**79/79** 27s 343ms/step - accuracy: 0.8914 - loss: 0.5837  
6 - val\_accuracy: 0.7097 - val\_loss: 1.0510  
Epoch 9/10  
**79/79** 0s 315ms/step - accuracy: 0.9039 - loss: 0.5405  
Epoch 9: val\_accuracy improved from 0.70973 to 0.80702, saving model to ./model2\_callbacks/model2\_chkpt/best\_model.weights.h5  
**79/79** 27s 343ms/step - accuracy: 0.9038 - loss: 0.5405  
7 - val\_accuracy: 0.8070 - val\_loss: 0.7715  
Epoch 10/10  
**79/79** 0s 316ms/step - accuracy: 0.9156 - loss: 0.5135  
Epoch 10: val\_accuracy did not improve from 0.80702  
**79/79** 27s 339ms/step - accuracy: 0.9155 - loss: 0.5135  
6 - val\_accuracy: 0.7959 - val\_loss: 0.9742

### Plotting Training and Validation Accuracy

In [34]: `plot_train_val_accuracy(regularized_fitted_model)`



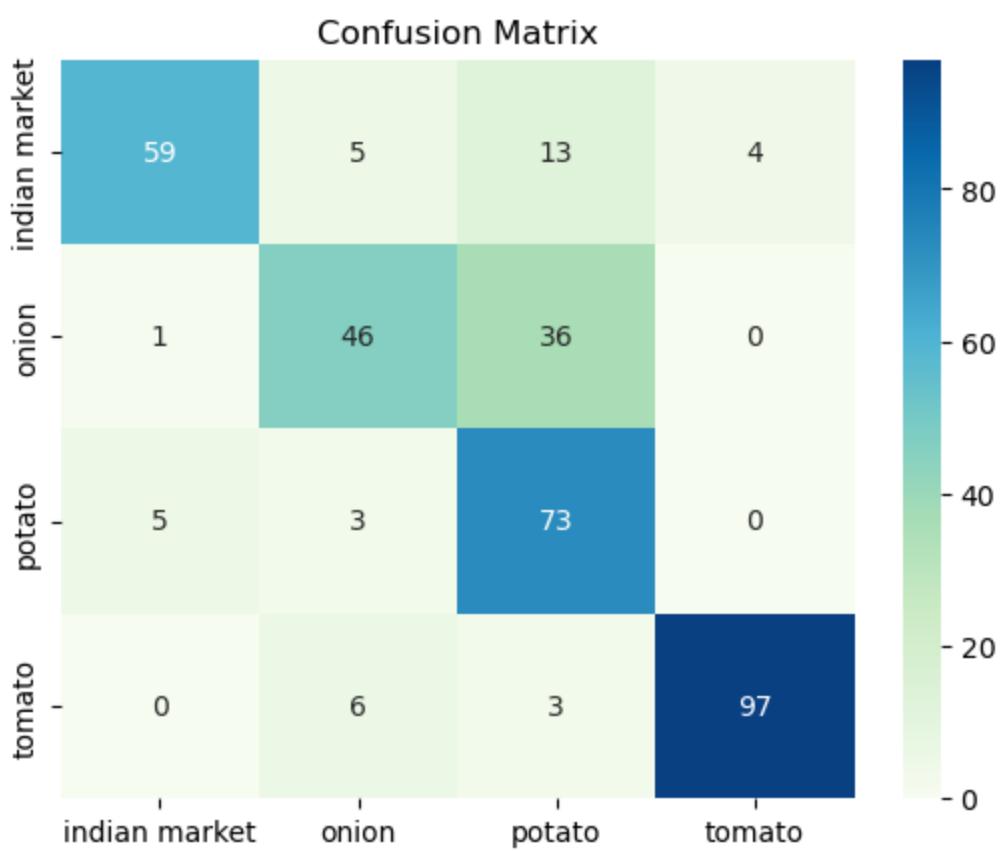
### Testing Accuracy

```
In [35]: model.load_weights(filepath=f'{direc}/model2_chkpt/best_model.weights.h5')
```

```
In [36]: basic_model_acc = print_accuracy(model,test_ds)
```

The accuracy of the model on the provided dataset is 78.35%

```
In [37]: print_confusion_matrix(model,test_ds,show_plot=True)
```



```
In [38]: built_models['regularized_model']=model
```

## Including Image Augmentation

Let's import some common augmentation layers from keras library to use it in our Training dataset.

```
In [39]: from tensorflow.keras.layers import RandomRotation, RandomFlip, RandomCrop, RandomContra
```

Implementing Augmentation on a sample image

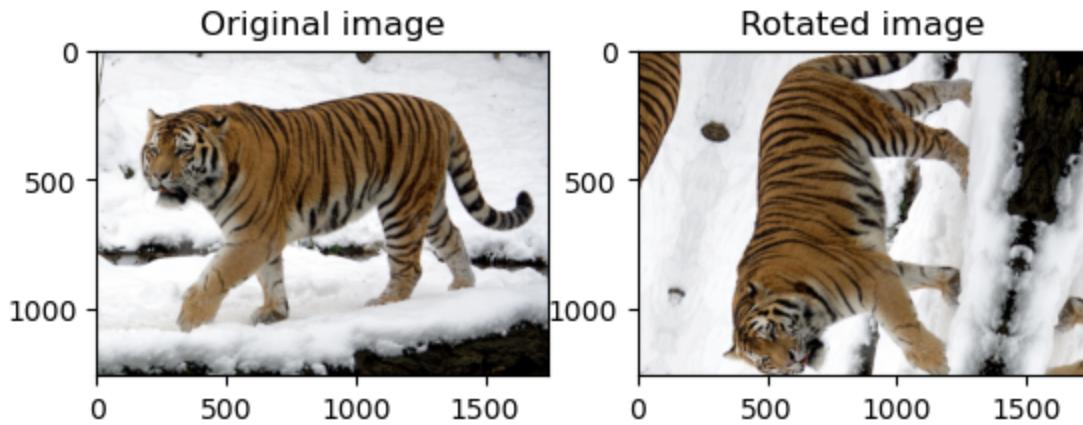
```
In [40]: img = tf.keras.preprocessing.image.load_img('./sample_image.jpg')
plt.figure(figsize=(5,5))
plt.imshow(img)
plt.show()
```



### Applying Random Rotation

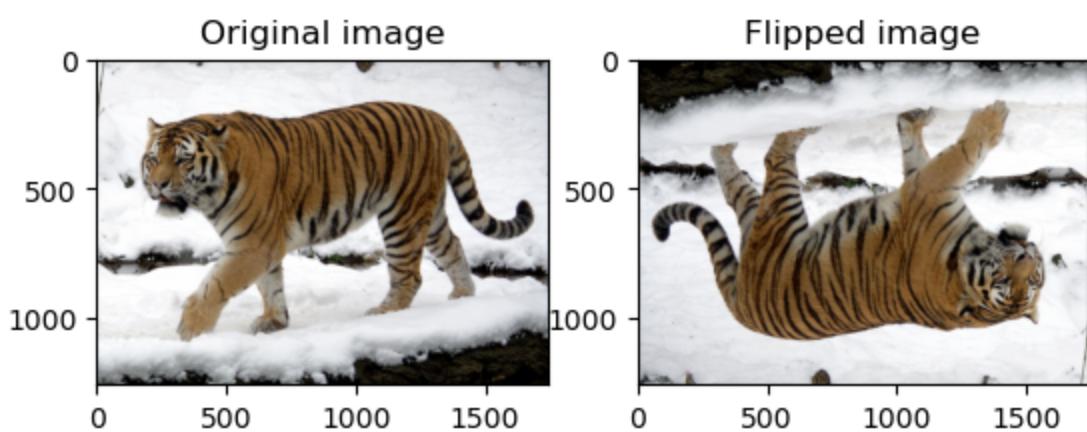
```
In [41]: def apply_augmentation(image,layer,operation):
    img_array = tf.keras.utils.img_to_array(image)
    plt.subplot(1,2,1)
    plt.title('Original image')
    plt.imshow(img_array.astype('uint8'))
    plt.subplot(1,2,2)
    plt.title(f'{operation} image')
    plt.imshow(layer(img_array).numpy().astype('uint8'))
    plt.show()
```

```
In [42]: rand_rotation = RandomRotation(factor=(-0.2, 0.3))
apply_augmentation(img,rand_rotation,'Rotated')
```



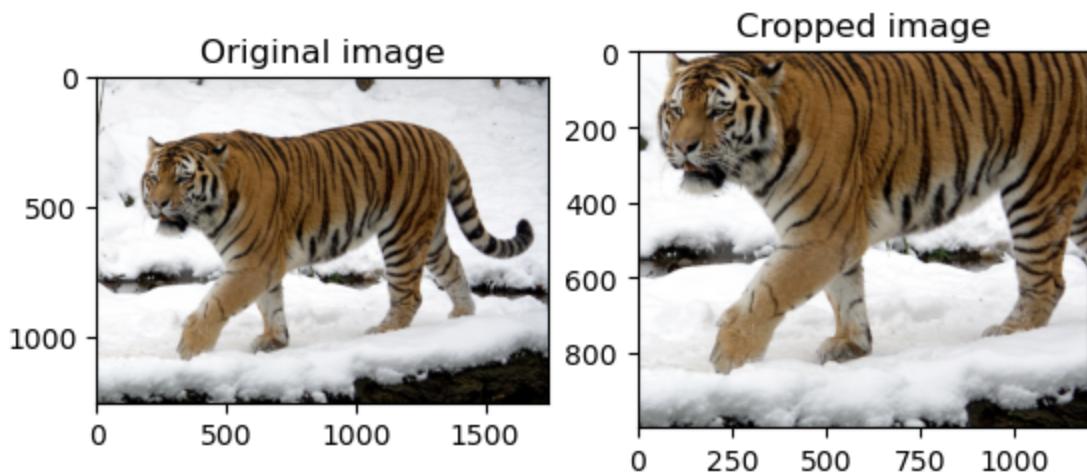
### Applying RandomFlip

```
In [43]: rand_flip = RandomFlip()
apply_augmentation(img,rand_flip,'Flipped')
```



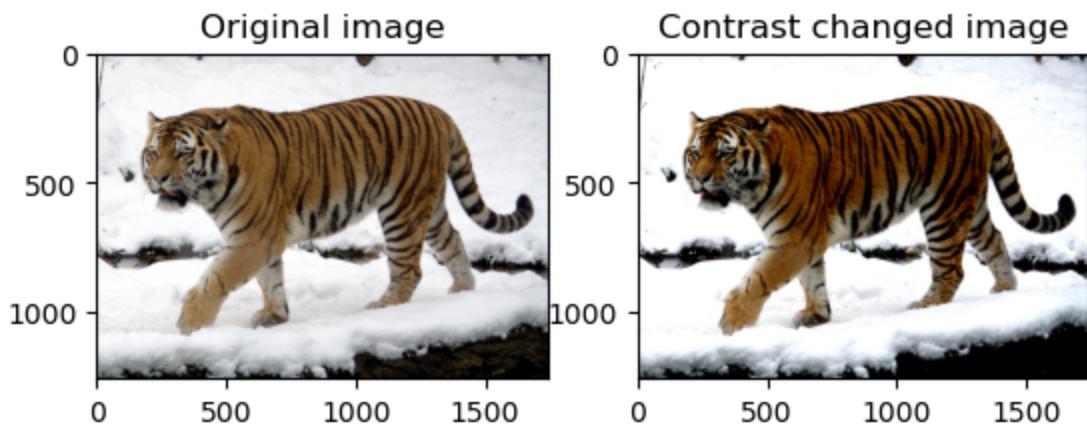
Applying Random Crop

```
In [44]: rand_crop = RandomCrop(height=1000,width=1200)  
apply_augmentation(img,rand_crop,'Cropped')
```



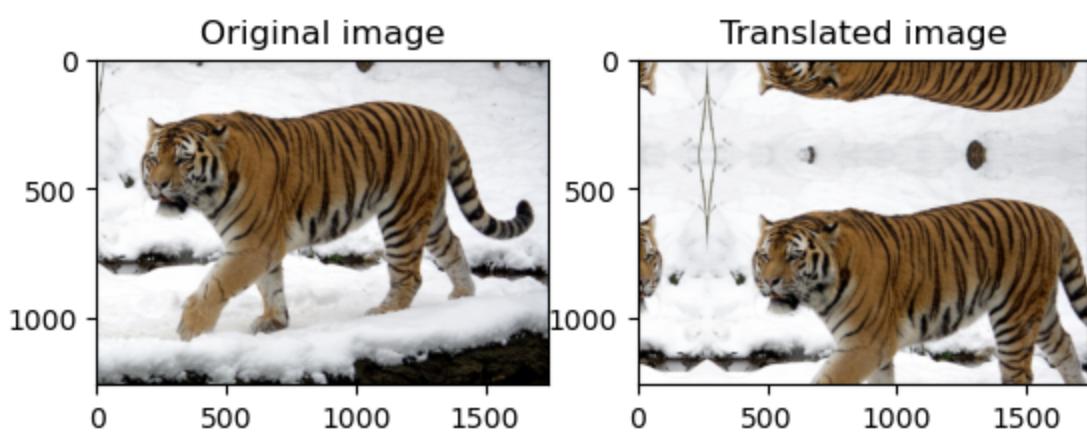
Applying Random Contrast

```
In [45]: rand_contrast = RandomContrast(factor=(0.2,0.7))  
apply_augmentation(img,rand_contrast,'Contrast changed')
```



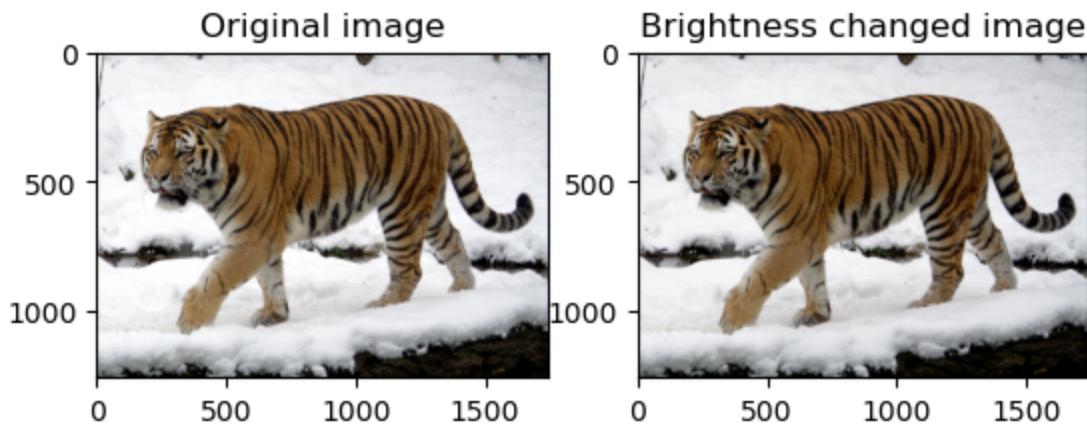
Applying Random Translation

```
In [46]: rand_translation = RandomTranslation(height_factor=(-0.2, 0.3),width_factor=(-0.2, 0.3))  
apply_augmentation(img,rand_translation,'Translated')
```



### Applying Random Brightness

```
In [47]: rand_brightness = RandomBrightness(factor=(-0.2, 0.8))
apply_augmentation(img, rand_brightness, 'Brightness changed')
```



### Applying Augmentation to the Training Dataset

```
In [48]: def apply_train_augmentation(data):      #We will apply Augmentation from Random Crop, Random Rotation, Random Brightness, Random Resizing
    augmentation = Sequential(
        name='training_augmentation',
        layers = [
            tf.keras.layers.Resizing(height=200, width=200),
            tf.keras.layers.RandomBrightness(factor=(-0.2, 0.7)),
            tf.keras.layers.RandomRotation(factor=(-0.2, 0.3)),
            tf.keras.layers.RandomCrop(height=128, width=128),
            tf.keras.layers.Rescaling(1.0/255)
        ]
    )

    train_ds = train_data.map(lambda x,y: (augmentation(x),y), num_parallel_calls=tf.data.AUTOTUNE)
    return train_ds
```

```
In [49]: aug_train_ds = apply_train_augmentation(train_data)
```

### Training the model on this new Training dataset

```
In [50]: aug_model = model_arch2()
aug_model.summary()
```

```
C:\Anaconda\lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning: Do not pass an `input_shape`/'input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "cnn\_model12"

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 126, 126, 16)	448
batch_normalization_6 (BatchNormalization)	(None, 126, 126, 16)	64
activation_7 (Activation)	(None, 126, 126, 16)	0
max_pooling2d_10 (MaxPooling2D)	(None, 63, 63, 16)	0
conv2d_11 (Conv2D)	(None, 61, 61, 32)	4,640
batch_normalization_7 (BatchNormalization)	(None, 61, 61, 32)	128
activation_8 (Activation)	(None, 61, 61, 32)	0
max_pooling2d_11 (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_12 (Conv2D)	(None, 28, 28, 64)	18,496
batch_normalization_8 (BatchNormalization)	(None, 28, 28, 64)	256
activation_9 (Activation)	(None, 28, 28, 64)	0
max_pooling2d_12 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_13 (Conv2D)	(None, 12, 12, 128)	73,856
batch_normalization_9 (BatchNormalization)	(None, 12, 12, 128)	512
activation_10 (Activation)	(None, 12, 12, 128)	0
max_pooling2d_13 (MaxPooling2D)	(None, 6, 6, 128)	0
conv2d_14 (Conv2D)	(None, 4, 4, 256)	295,168
batch_normalization_10 (BatchNormalization)	(None, 4, 4, 256)	1,024
activation_11 (Activation)	(None, 4, 4, 256)	0
max_pooling2d_14 (MaxPooling2D)	(None, 2, 2, 256)	0
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 256)	0
dense_5 (Dense)	(None, 128)	32,896
batch_normalization_11 (BatchNormalization)	(None, 128)	512
activation_12 (Activation)	(None, 128)	0
activation_13 (Activation)	(None, 128)	0

dense\_6 (Dense)

(None, 4)

516

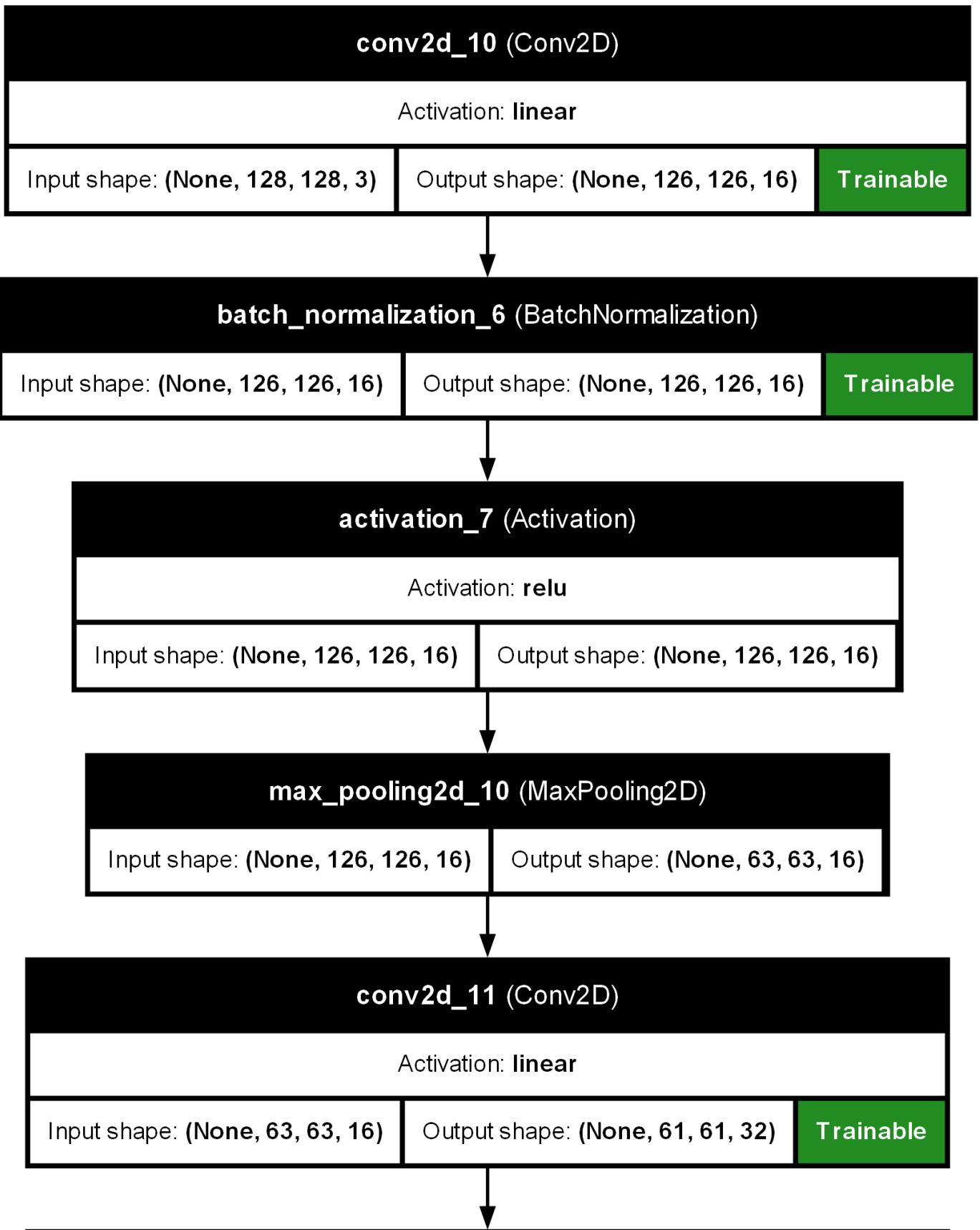
Total params: 428,516 (1.63 MB)  
Trainable params: 427,268 (1.63 MB)  
Non-trainable params: 1,248 (4.88 KB)

Plotting the model

In [156...]

```
tf.keras.utils.plot_model(aug_model, show_shapes=True, show_layer_activations=True, show_la
```

Out[156]:



## batch\_normalization\_7 (BatchNormalization)

Input shape: (None, 61, 61, 32)

Output shape: (None, 61, 61, 32)

Trainable

## activation\_8 (Activation)

Activation: relu

Input shape: (None, 61, 61, 32)

Output shape: (None, 61, 61, 32)

## max\_pooling2d\_11 (MaxPooling2D)

Input shape: (None, 61, 61, 32)

Output shape: (None, 30, 30, 32)

## conv2d\_12 (Conv2D)

Activation: linear

Input shape: (None, 30, 30, 32)

Output shape: (None, 28, 28, 64)

Trainable

## batch\_normalization\_8 (BatchNormalization)

Input shape: (None, 28, 28, 64)

Output shape: (None, 28, 28, 64)

Trainable

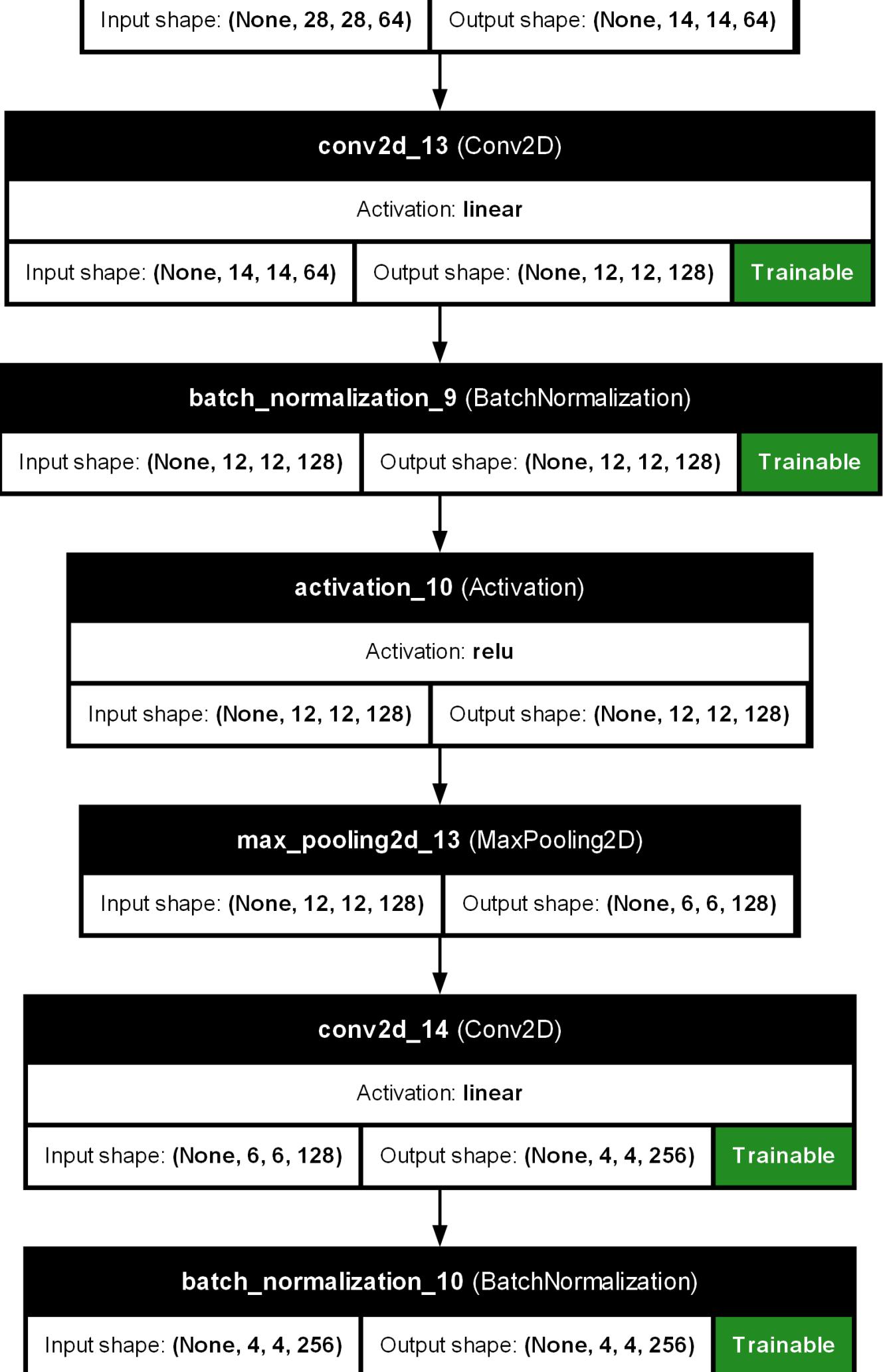
## activation\_9 (Activation)

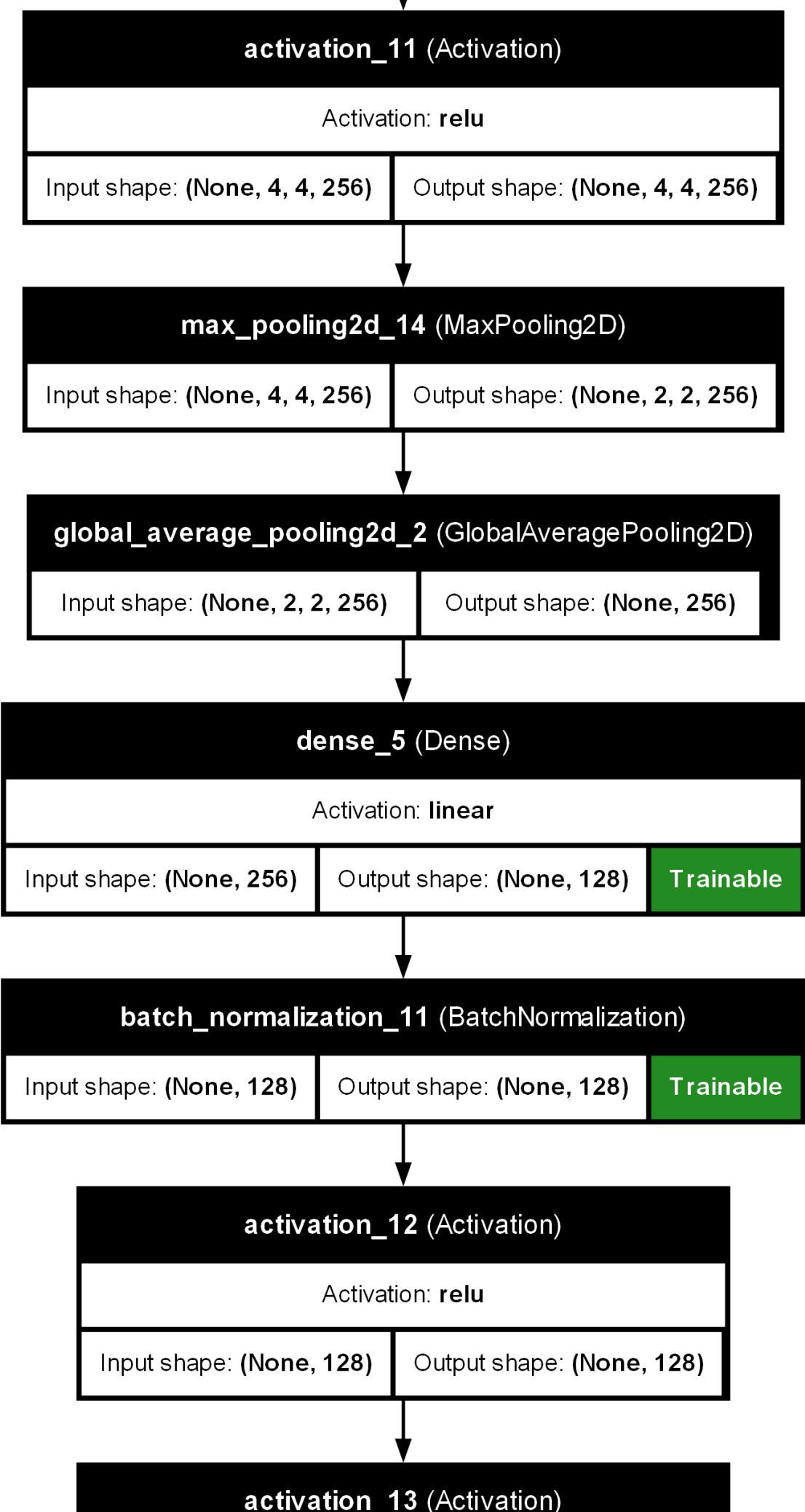
Activation: relu

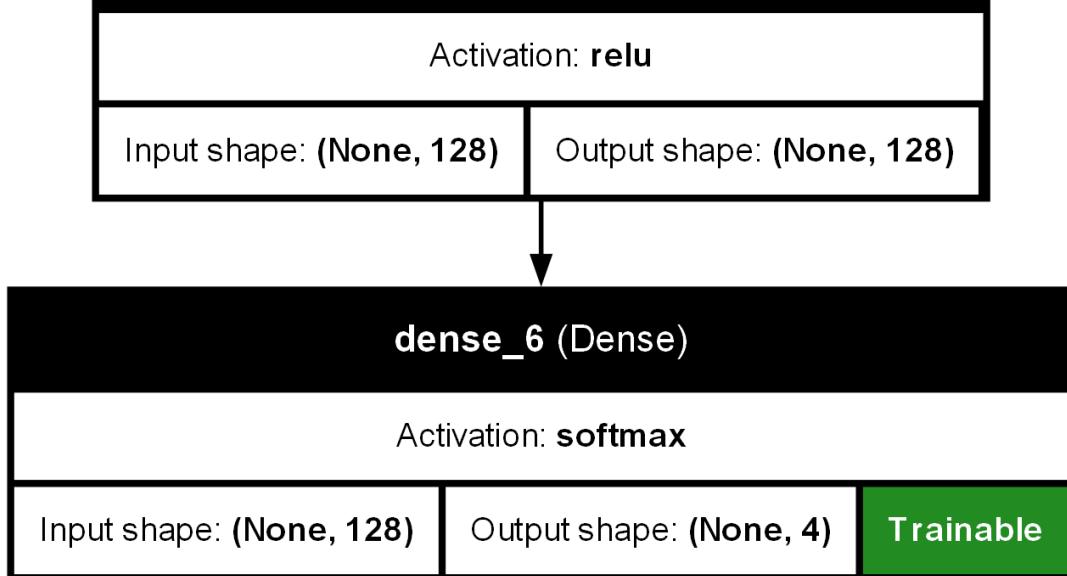
Input shape: (None, 28, 28, 64)

Output shape: (None, 28, 28, 64)

## max\_pooling2d\_12 (MaxPooling2D)







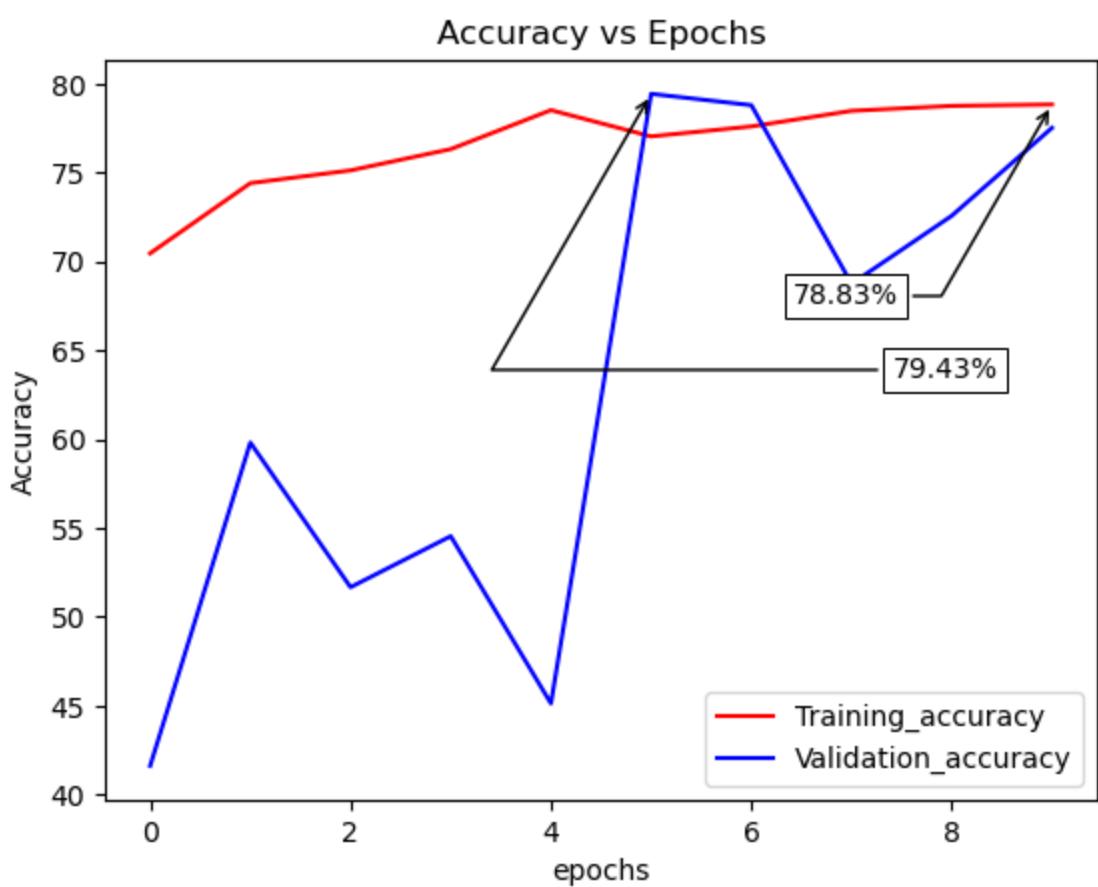
Compiling the model for augmented data

```
In [51]: aug_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
directory = './aug_model_callbacks'
aug_model_chkpt = tf.keras.callbacks.ModelCheckpoint(f'{directory}/aug_model_chkpt/best_'
aug_model_tensorboard = tf.keras.callbacks.TensorBoard(log_dir=f'{directory}/aug_model_t'
aug_model_earlystop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10, mi
aug_callbacks = [aug_model_chkpt, aug_model_tensorboard, aug_model_earlystop]
aug_fitted_model = aug_model.fit(aug_train_ds, validation_data=validation_ds, batch_size=3

Epoch 1/10
79/79 39s 368ms/step - accuracy: 0.6592 - loss: 1.356
9 - val_accuracy: 0.4163 - val_loss: 1.9275
Epoch 2/10
79/79 29s 356ms/step - accuracy: 0.7432 - loss: 1.150
7 - val_accuracy: 0.5981 - val_loss: 1.6994
Epoch 3/10
79/79 29s 357ms/step - accuracy: 0.7434 - loss: 1.084
0 - val_accuracy: 0.5167 - val_loss: 1.7463
Epoch 4/10
79/79 29s 356ms/step - accuracy: 0.7768 - loss: 0.978
7 - val_accuracy: 0.5455 - val_loss: 1.4505
Epoch 5/10
79/79 1289s 17s/step - accuracy: 0.7875 - loss: 0.921
6 - val_accuracy: 0.4514 - val_loss: 2.3326
Epoch 6/10
79/79 40s 494ms/step - accuracy: 0.7699 - loss: 0.914
1 - val_accuracy: 0.7943 - val_loss: 0.8330
Epoch 7/10
79/79 38s 481ms/step - accuracy: 0.7768 - loss: 0.868
2 - val_accuracy: 0.7879 - val_loss: 0.8079
Epoch 8/10
79/79 35s 431ms/step - accuracy: 0.8007 - loss: 0.809
5 - val_accuracy: 0.6874 - val_loss: 1.0330
Epoch 9/10
79/79 31s 392ms/step - accuracy: 0.7946 - loss: 0.771
4 - val_accuracy: 0.7257 - val_loss: 1.0862
Epoch 10/10
79/79 33s 412ms/step - accuracy: 0.7886 - loss: 0.787
6 - val_accuracy: 0.7751 - val_loss: 0.7492
```

Plotting Training and Validation accuracy

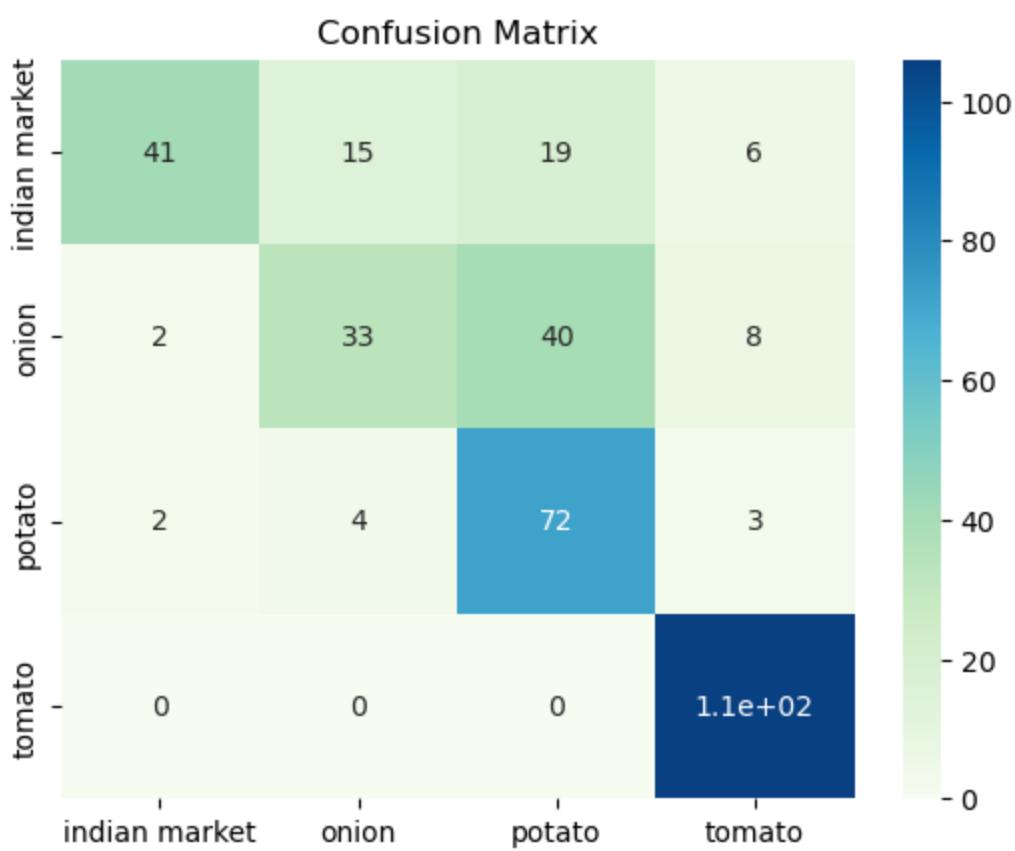
```
In [52]: plot_train_val_accuracy(aug_fitted_model)
```



#### Metrics on Test Data

```
In [53]: aug_model.load_weights(f'{directory}/aug_model_chkpt/best_model.weights.h5')
print_accuracy(aug_model,test_ds)
print()
print_confusion_matrix(aug_model,test_ds,show_plot=True)
```

The accuracy of the model on the provided dataset is 71.79%



As we have limited data to train the CNN model, we are not able to get good testing accuracy with the number of epochs equal to 10. In cases where we dont have enough training data, we can use pretrained Transfer learning models to classify our images into different classes.

```
In [54]: built_models['Augmented_model']=aug_model
```

## Classification using VGG 16 Model

We will use the Pre-trained VGG Net 16 model with the "Imagenet" weights and include top set as False, which will cutt off the dense layers included in the model

```
In [55]: vgg_model = tf.keras.applications.VGG16(weights="imagenet", include_top=False, input_shape=vgg_model.trainable=False)
```

```
In [56]: vgg_model.summary()
```

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_layer_5 (InputLayer)	(None, 128, 128, 3)	0
block1_conv1 (Conv2D)	(None, 128, 128, 64)	1,792
block1_conv2 (Conv2D)	(None, 128, 128, 64)	36,928
block1_pool (MaxPooling2D)	(None, 64, 64, 64)	0
block2_conv1 (Conv2D)	(None, 64, 64, 128)	73,856
block2_conv2 (Conv2D)	(None, 64, 64, 128)	147,584

block2_pool (MaxPooling2D)	(None, 32, 32, 128)	0
block3_conv1 (Conv2D)	(None, 32, 32, 256)	295,168
block3_conv2 (Conv2D)	(None, 32, 32, 256)	590,080
block3_conv3 (Conv2D)	(None, 32, 32, 256)	590,080
block3_pool (MaxPooling2D)	(None, 16, 16, 256)	0
block4_conv1 (Conv2D)	(None, 16, 16, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 16, 16, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 16, 16, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 8, 8, 512)	0
block5_conv1 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 8, 8, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 512)	0

Total params: 14,714,688 (56.13 MB)

Trainable params: 0 (0.00 B)

Non-trainable params: 14,714,688 (56.13 MB)

```
In [57]: vgg_inc_model = tf.keras.models.Sequential(
    name="vgg_included_model",
    layers = [
        vgg_model,
        tf.keras.layers.Dense(units=512, activation='relu', use_bias=True),
        tf.keras.layers.Dense(units=4, activation='softmax', use_bias=True)      #Used to cl
    ]
)
```

```
In [58]: vgg_inc_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
vgg_inc_model.summary()
```

Model: "vgg\_included\_model"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	?	14,714,688
dense_7 (Dense)	?	0 (unbuilt)
dense_8 (Dense)	?	0 (unbuilt)

Total params: 14,714,688 (56.13 MB)

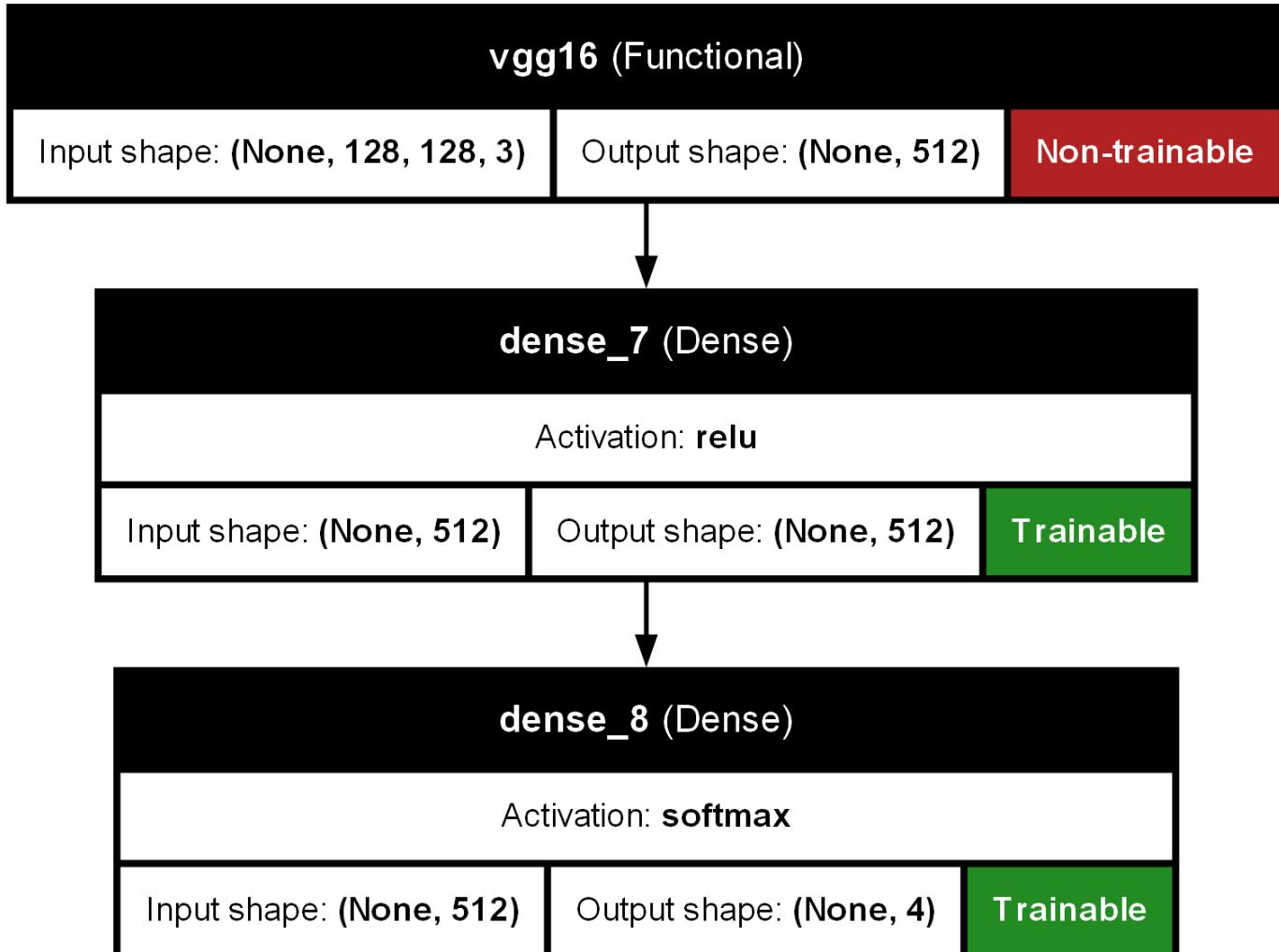
Trainable params: 0 (0.00 B)

Non-trainable params: 14,714,688 (56.13 MB)

## Plotting the model

```
In [155]: tf.keras.utils.plot_model(vgg_inc_model, show_shapes=True, show_layer_names=True, show_laye
```

Out[155]:



We will Train this VGG included model on the augmented dataset

```
In [59]: directory = './vgg_model_callbacks'
vgg_model_chkpt = tf.keras.callbacks.ModelCheckpoint(f'{directory}/vgg_model_chkpt/best_'
vgg_model_tensorboard = tf.keras.callbacks.TensorBoard(log_dir=f'{directory}/vgg_model_t'
vgg_model_earlystop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10, mi
vgg_callbacks = [vgg_model_chkpt, vgg_model_tensorboard, vgg_model_earlystop]
vgg_fitted_model = vgg_inc_model.fit(aug_train_ds, validation_data=validation_ds, batch_si
```

Epoch 1/10  
79/79 212s 3s/step - accuracy: 0.5844 - loss: 0.9864  
- val\_accuracy: 0.7400 - val\_loss: 0.6232

Epoch 2/10  
79/79 231s 3s/step - accuracy: 0.7562 - loss: 0.5966  
- val\_accuracy: 0.8246 - val\_loss: 0.4405

Epoch 3/10  
79/79 227s 3s/step - accuracy: 0.7807 - loss: 0.5445  
- val\_accuracy: 0.8325 - val\_loss: 0.4264

Epoch 4/10  
79/79 226s 3s/step - accuracy: 0.7964 - loss: 0.5052  
- val\_accuracy: 0.8309 - val\_loss: 0.4414

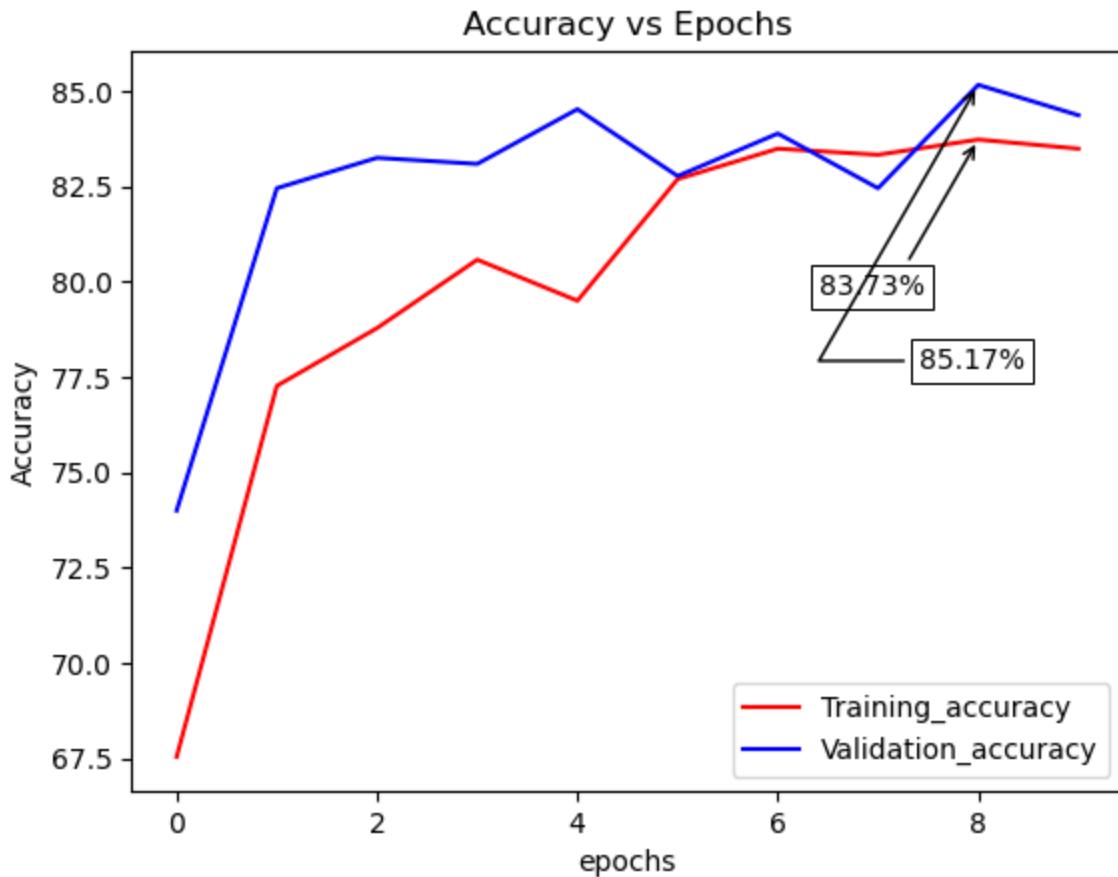
Epoch 5/10  
79/79 226s 3s/step - accuracy: 0.7751 - loss: 0.5153  
- val\_accuracy: 0.8453 - val\_loss: 0.3858

Epoch 6/10  
79/79 226s 3s/step - accuracy: 0.8353 - loss: 0.4270  
- val\_accuracy: 0.8278 - val\_loss: 0.4150

```
Epoch 7/10
79/79 226s 3s/step - accuracy: 0.8393 - loss: 0.3949
- val_accuracy: 0.8389 - val_loss: 0.3921
Epoch 8/10
79/79 227s 3s/step - accuracy: 0.8428 - loss: 0.4052
- val_accuracy: 0.8246 - val_loss: 0.4617
Epoch 9/10
79/79 225s 3s/step - accuracy: 0.8319 - loss: 0.4099
- val_accuracy: 0.8517 - val_loss: 0.3843
Epoch 10/10
79/79 225s 3s/step - accuracy: 0.8379 - loss: 0.4323
- val_accuracy: 0.8437 - val_loss: 0.4023
```

### Training and Validation Accuracy

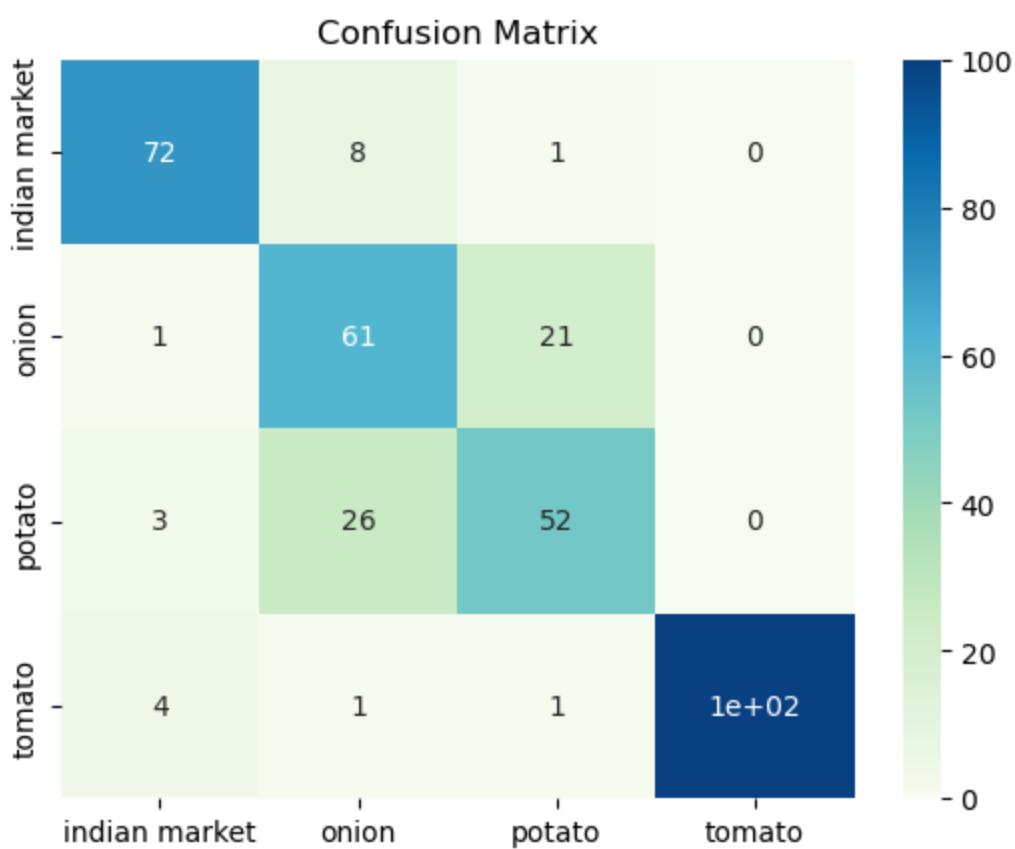
```
In [60]: plot_train_val_accuracy(vgg_fitted_model)
```



### Testing Accuracy

```
In [61]: vgg_inc_model.load_weights(filepath=f'{directory}/vgg_model_chkpt/best_model.weights.h5'
print_accuracy(vgg_inc_model,test_ds)
print()
print_confusion_matrix(vgg_inc_model,test_ds,show_plot=True)
```

The accuracy of the model on the provided dataset is 81.20%



```
In [62]: built_models['vgg_model'] = vgg_inc_model
```

## Implementing CNN model using Resnet

```
In [63]: from tensorflow.keras.applications import ResNet50
```

```
In [64]: resnet_model = ResNet50(include_top=False, weights='imagenet', input_shape=(128,128,3), pooling='max')
resnet_model.trainable=False
resnet_model.summary()
```

Model: "resnet50"

Layer (type)	Output Shape	Param #	Connected To
input_layer_7 (InputLayer)	(None, 128, 128, 3)	0	-
conv1_pad (ZeroPadding2D)	(None, 134, 134, 3)	0	input_layer_7
conv1_conv (Conv2D)	(None, 64, 64, 64)	9,472	conv1_pad[0]
conv1_bn (BatchNormalization)	(None, 64, 64, 64)	256	conv1_conv[0]
conv1_relu (Activation)	(None, 64, 64, 64)	0	conv1_bn[0]
pool1_pad (ZeroPadding2D)	(None, 66, 66, 64)	0	conv1_relu[0]
pool1_pool (MaxPooling2D)	(None, 32, 32, 64)	0	pool1_pad[0]
conv2_block1_1_conv (Conv2D)	(None, 32, 32, 64)	4,160	pool1_pool[0]
conv2_block1_1_bn (BatchNormalization)	(None, 32, 32, 64)	256	conv2_block1_1_conv[0]
conv2_block1_1_relu	(None, 32, 32, 64)	0	conv2_block1_1_bn[0]

	(Activation)			
conv2_block1_2_conv (Conv2D)	(None, 32, 32, 64)	36,928	conv2_bloc	
conv2_block1_2_bn (BatchNormalization)	(None, 32, 32, 64)	256	conv2_bloc	
conv2_block1_2_relu (Activation)	(None, 32, 32, 64)	0	conv2_bloc	
conv2_block1_0_conv (Conv2D)	(None, 32, 32, 256)	16,640	pool1_pool	
conv2_block1_3_conv (Conv2D)	(None, 32, 32, 256)	16,640	conv2_bloc	
conv2_block1_0_bn (BatchNormalization)	(None, 32, 32, 256)	1,024	conv2_bloc	
conv2_block1_3_bn (BatchNormalization)	(None, 32, 32, 256)	1,024	conv2_bloc	
conv2_block1_add (Add)	(None, 32, 32, 256)	0	conv2_bloc	conv2_bloc
conv2_block1_out (Activation)	(None, 32, 32, 256)	0	conv2_bloc	
conv2_block2_1_conv (Conv2D)	(None, 32, 32, 64)	16,448	conv2_bloc	
conv2_block2_1_bn (BatchNormalization)	(None, 32, 32, 64)	256	conv2_bloc	
conv2_block2_1_relu (Activation)	(None, 32, 32, 64)	0	conv2_bloc	
conv2_block2_2_conv (Conv2D)	(None, 32, 32, 64)	36,928	conv2_bloc	
conv2_block2_2_bn (BatchNormalization)	(None, 32, 32, 64)	256	conv2_bloc	
conv2_block2_2_relu (Activation)	(None, 32, 32, 64)	0	conv2_bloc	
conv2_block2_3_conv (Conv2D)	(None, 32, 32, 256)	16,640	conv2_bloc	
conv2_block2_3_bn (BatchNormalization)	(None, 32, 32, 256)	1,024	conv2_bloc	
conv2_block2_add (Add)	(None, 32, 32, 256)	0	conv2_bloc	conv2_bloc
conv2_block2_out (Activation)	(None, 32, 32, 256)	0	conv2_bloc	
conv2_block3_1_conv (Conv2D)	(None, 32, 32, 64)	16,448	conv2_bloc	
conv2_block3_1_bn (BatchNormalization)	(None, 32, 32, 64)	256	conv2_bloc	
conv2_block3_1_relu (Activation)	(None, 32, 32, 64)	0	conv2_bloc	
conv2_block3_2_conv (Conv2D)	(None, 32, 32, 64)	36,928	conv2_bloc	
conv2_block3_2_bn	(None, 32, 32, 64)	256	conv2_bloc	

(BatchNormalization)			
conv2_block3_2_relu (Activation)	(None, 32, 32, 64)	0	conv2_bloc
conv2_block3_3_conv (Conv2D)	(None, 32, 32, 256)	16,640	conv2_bloc
conv2_block3_3_bn (BatchNormalization)	(None, 32, 32, 256)	1,024	conv2_bloc
conv2_block3_add (Add)	(None, 32, 32, 256)	0	conv2_bloc
conv2_block3_out (Activation)	(None, 32, 32, 256)	0	conv2_bloc
conv3_block1_1_conv (Conv2D)	(None, 16, 16, 128)	32,896	conv2_bloc
conv3_block1_1_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block1_1_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc
conv3_block1_2_conv (Conv2D)	(None, 16, 16, 128)	147,584	conv3_bloc
conv3_block1_2_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block1_2_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc
conv3_block1_0_conv (Conv2D)	(None, 16, 16, 512)	131,584	conv2_bloc
conv3_block1_3_conv (Conv2D)	(None, 16, 16, 512)	66,048	conv3_bloc
conv3_block1_0_bn (BatchNormalization)	(None, 16, 16, 512)	2,048	conv3_bloc
conv3_block1_3_bn (BatchNormalization)	(None, 16, 16, 512)	2,048	conv3_bloc
conv3_block1_add (Add)	(None, 16, 16, 512)	0	conv3_bloc
conv3_block1_out (Activation)	(None, 16, 16, 512)	0	conv3_bloc
conv3_block2_1_conv (Conv2D)	(None, 16, 16, 128)	65,664	conv3_bloc
conv3_block2_1_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block2_1_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc
conv3_block2_2_conv (Conv2D)	(None, 16, 16, 128)	147,584	conv3_bloc
conv3_block2_2_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block2_2_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc

conv3_block2_3_conv (Conv2D)	(None, 16, 16, 512)	66,048	conv3_bloc
conv3_block2_3_bn (BatchNormalization)	(None, 16, 16, 512)	2,048	conv3_bloc
conv3_block2_add (Add)	(None, 16, 16, 512)	0	conv3_bloc conv3_bloc
conv3_block2_out (Activation)	(None, 16, 16, 512)	0	conv3_bloc
conv3_block3_1_conv (Conv2D)	(None, 16, 16, 128)	65,664	conv3_bloc
conv3_block3_1_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block3_1_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc
conv3_block3_2_conv (Conv2D)	(None, 16, 16, 128)	147,584	conv3_bloc
conv3_block3_2_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block3_2_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc
conv3_block3_3_conv (Conv2D)	(None, 16, 16, 512)	66,048	conv3_bloc
conv3_block3_3_bn (BatchNormalization)	(None, 16, 16, 512)	2,048	conv3_bloc
conv3_block3_add (Add)	(None, 16, 16, 512)	0	conv3_bloc conv3_bloc
conv3_block3_out (Activation)	(None, 16, 16, 512)	0	conv3_bloc
conv3_block4_1_conv (Conv2D)	(None, 16, 16, 128)	65,664	conv3_bloc
conv3_block4_1_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block4_1_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc
conv3_block4_2_conv (Conv2D)	(None, 16, 16, 128)	147,584	conv3_bloc
conv3_block4_2_bn (BatchNormalization)	(None, 16, 16, 128)	512	conv3_bloc
conv3_block4_2_relu (Activation)	(None, 16, 16, 128)	0	conv3_bloc
conv3_block4_3_conv (Conv2D)	(None, 16, 16, 512)	66,048	conv3_bloc
conv3_block4_3_bn (BatchNormalization)	(None, 16, 16, 512)	2,048	conv3_bloc
conv3_block4_add (Add)	(None, 16, 16, 512)	0	conv3_bloc conv3_bloc
conv3_block4_out (Activation)	(None, 16, 16, 512)	0	conv3_bloc

conv4_block1_1_conv (Conv2D)	(None, 8, 8, 256)	131,328	conv3_bloc
conv4_block1_1_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block1_1_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block1_2_conv (Conv2D)	(None, 8, 8, 256)	590,080	conv4_bloc
conv4_block1_2_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block1_2_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block1_0_conv (Conv2D)	(None, 8, 8, 1024)	525,312	conv3_bloc
conv4_block1_3_conv (Conv2D)	(None, 8, 8, 1024)	263,168	conv4_bloc
conv4_block1_0_bn (BatchNormalization)	(None, 8, 8, 1024)	4,096	conv4_bloc
conv4_block1_3_bn (BatchNormalization)	(None, 8, 8, 1024)	4,096	conv4_bloc
conv4_block1_add (Add)	(None, 8, 8, 1024)	0	conv4_bloc conv4_bloc
conv4_block1_out (Activation)	(None, 8, 8, 1024)	0	conv4_bloc
conv4_block2_1_conv (Conv2D)	(None, 8, 8, 256)	262,400	conv4_bloc
conv4_block2_1_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block2_1_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block2_2_conv (Conv2D)	(None, 8, 8, 256)	590,080	conv4_bloc
conv4_block2_2_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block2_2_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block2_3_conv (Conv2D)	(None, 8, 8, 1024)	263,168	conv4_bloc
conv4_block2_3_bn (BatchNormalization)	(None, 8, 8, 1024)	4,096	conv4_bloc
conv4_block2_add (Add)	(None, 8, 8, 1024)	0	conv4_bloc conv4_bloc
conv4_block2_out (Activation)	(None, 8, 8, 1024)	0	conv4_bloc
conv4_block3_1_conv (Conv2D)	(None, 8, 8, 256)	262,400	conv4_bloc
conv4_block3_1_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc

conv4_block3_1_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block3_2_conv (Conv2D)	(None, 8, 8, 256)	590,080	conv4_bloc
conv4_block3_2_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block3_2_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block3_3_conv (Conv2D)	(None, 8, 8, 1024)	263,168	conv4_bloc
conv4_block3_3_bn (BatchNormalization)	(None, 8, 8, 1024)	4,096	conv4_bloc
conv4_block3_add (Add)	(None, 8, 8, 1024)	0	conv4_bloc conv4_bloc
conv4_block3_out (Activation)	(None, 8, 8, 1024)	0	conv4_bloc
conv4_block4_1_conv (Conv2D)	(None, 8, 8, 256)	262,400	conv4_bloc
conv4_block4_1_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block4_1_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block4_2_conv (Conv2D)	(None, 8, 8, 256)	590,080	conv4_bloc
conv4_block4_2_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block4_2_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block4_3_conv (Conv2D)	(None, 8, 8, 1024)	263,168	conv4_bloc
conv4_block4_3_bn (BatchNormalization)	(None, 8, 8, 1024)	4,096	conv4_bloc
conv4_block4_add (Add)	(None, 8, 8, 1024)	0	conv4_bloc conv4_bloc
conv4_block4_out (Activation)	(None, 8, 8, 1024)	0	conv4_bloc
conv4_block5_1_conv (Conv2D)	(None, 8, 8, 256)	262,400	conv4_bloc
conv4_block5_1_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block5_1_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block5_2_conv (Conv2D)	(None, 8, 8, 256)	590,080	conv4_bloc
conv4_block5_2_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block5_2_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc

conv4_block5_3_conv (Conv2D)	(None, 8, 8, 1024)	263,168	conv4_bloc
conv4_block5_3_bn (BatchNormalization)	(None, 8, 8, 1024)	4,096	conv4_bloc
conv4_block5_add (Add)	(None, 8, 8, 1024)	0	conv4_bloc
conv4_block5_out (Activation)	(None, 8, 8, 1024)	0	conv4_bloc
conv4_block6_1_conv (Conv2D)	(None, 8, 8, 256)	262,400	conv4_bloc
conv4_block6_1_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block6_1_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block6_2_conv (Conv2D)	(None, 8, 8, 256)	590,080	conv4_bloc
conv4_block6_2_bn (BatchNormalization)	(None, 8, 8, 256)	1,024	conv4_bloc
conv4_block6_2_relu (Activation)	(None, 8, 8, 256)	0	conv4_bloc
conv4_block6_3_conv (Conv2D)	(None, 8, 8, 1024)	263,168	conv4_bloc
conv4_block6_3_bn (BatchNormalization)	(None, 8, 8, 1024)	4,096	conv4_bloc
conv4_block6_add (Add)	(None, 8, 8, 1024)	0	conv4_bloc
conv4_block6_out (Activation)	(None, 8, 8, 1024)	0	conv4_bloc
conv5_block1_1_conv (Conv2D)	(None, 4, 4, 512)	524,800	conv4_bloc
conv5_block1_1_bn (BatchNormalization)	(None, 4, 4, 512)	2,048	conv5_bloc
conv5_block1_1_relu (Activation)	(None, 4, 4, 512)	0	conv5_bloc
conv5_block1_2_conv (Conv2D)	(None, 4, 4, 512)	2,359,808	conv5_bloc
conv5_block1_2_bn (BatchNormalization)	(None, 4, 4, 512)	2,048	conv5_bloc
conv5_block1_2_relu (Activation)	(None, 4, 4, 512)	0	conv5_bloc
conv5_block1_0_conv (Conv2D)	(None, 4, 4, 2048)	2,099,200	conv4_bloc
conv5_block1_3_conv (Conv2D)	(None, 4, 4, 2048)	1,050,624	conv5_bloc
conv5_block1_0_bn (BatchNormalization)	(None, 4, 4, 2048)	8,192	conv5_bloc
conv5_block1_3_bn (BatchNormalization)	(None, 4, 4, 2048)	8,192	conv5_bloc

conv5_block1_add (Add)	(None, 4, 4, 2048)	0	conv5_bloc conv5_bloc
conv5_block1_out (Activation)	(None, 4, 4, 2048)	0	conv5_bloc
conv5_block2_1_conv (Conv2D)	(None, 4, 4, 512)	1,049,088	conv5_bloc
conv5_block2_1_bn (BatchNormalization)	(None, 4, 4, 512)	2,048	conv5_bloc
conv5_block2_1_relu (Activation)	(None, 4, 4, 512)	0	conv5_bloc
conv5_block2_2_conv (Conv2D)	(None, 4, 4, 512)	2,359,808	conv5_bloc
conv5_block2_2_bn (BatchNormalization)	(None, 4, 4, 512)	2,048	conv5_bloc
conv5_block2_2_relu (Activation)	(None, 4, 4, 512)	0	conv5_bloc
conv5_block2_3_conv (Conv2D)	(None, 4, 4, 2048)	1,050,624	conv5_bloc
conv5_block2_3_bn (BatchNormalization)	(None, 4, 4, 2048)	8,192	conv5_bloc
conv5_block2_add (Add)	(None, 4, 4, 2048)	0	conv5_bloc conv5_bloc
conv5_block2_out (Activation)	(None, 4, 4, 2048)	0	conv5_bloc
conv5_block3_1_conv (Conv2D)	(None, 4, 4, 512)	1,049,088	conv5_bloc
conv5_block3_1_bn (BatchNormalization)	(None, 4, 4, 512)	2,048	conv5_bloc
conv5_block3_1_relu (Activation)	(None, 4, 4, 512)	0	conv5_bloc
conv5_block3_2_conv (Conv2D)	(None, 4, 4, 512)	2,359,808	conv5_bloc
conv5_block3_2_bn (BatchNormalization)	(None, 4, 4, 512)	2,048	conv5_bloc
conv5_block3_2_relu (Activation)	(None, 4, 4, 512)	0	conv5_bloc
conv5_block3_3_conv (Conv2D)	(None, 4, 4, 2048)	1,050,624	conv5_bloc
conv5_block3_3_bn (BatchNormalization)	(None, 4, 4, 2048)	8,192	conv5_bloc
conv5_block3_add (Add)	(None, 4, 4, 2048)	0	conv5_bloc conv5_bloc
conv5_block3_out (Activation)	(None, 4, 4, 2048)	0	conv5_bloc
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	conv5_bloc

Total params: 23,587,712 (89.98 MB)

Trainable params: 23,534,592 (89.78 MB)

Non-trainable params: 53,120 (207.50 KB)

From the summary, it can be seen that there are about 23,534,592 Trainable parameters in the RestNet 50 model

```
In [65]: resnet_inc_model = tf.keras.models.Sequential(  
    name='Resnet_model',  
    layers = [  
        resnet_model,  
        tf.keras.layers.Dense(units=2048, activation='relu', use_bias=True),  
        tf.keras.layers.Dense(units=4, activation='softmax', use_bias=True)  
    ]  
)  
resnet_inc_model.summary()
```

Model: "Resnet\_model"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	?	23,587,712
dense_9 (Dense)	?	0 (unbuilt)
dense_10 (Dense)	?	0 (unbuilt)

Total params: 23,587,712 (89.98 MB)

Trainable params: 23,534,592 (89.78 MB)

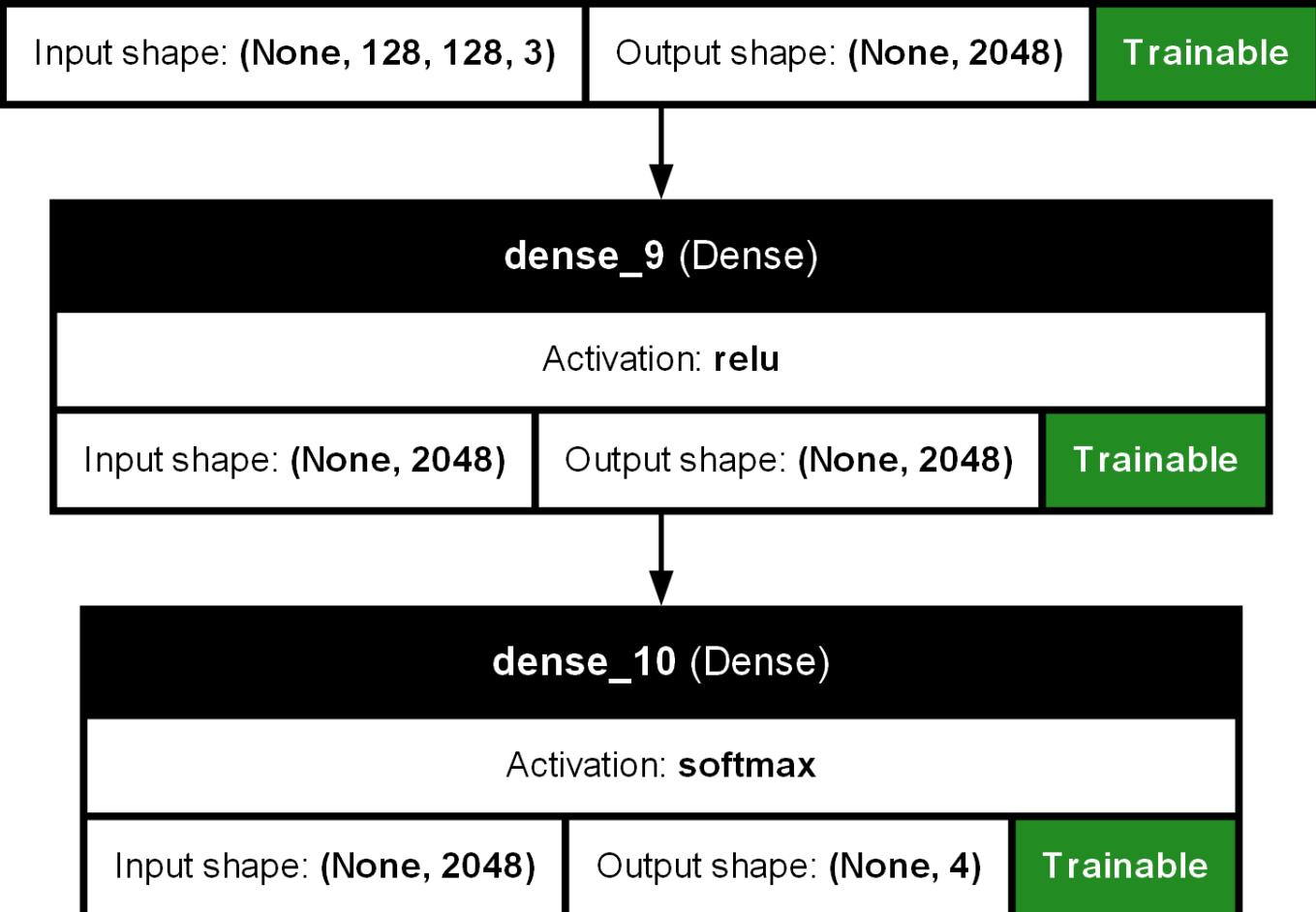
Non-trainable params: 53,120 (207.50 KB)

Plotting the model

```
In [154... tf.keras.utils.plot_model(resnet_inc_model, show_shapes=True, show_layer_names=True, show_l
```

Out[154]:

## resnet50 (Functional)



Training the model on Augmented Training dataset ---- Setting the number of epochs as 5 as there are more Trainable parameters.

```
In [66]: resnet_inc_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
directory=f'./resnet_callbacks'
resnet_modelchkpt = tf.keras.callbacks.ModelCheckpoint(filepath=f'{directory}/resnet_model.h5')
resnet_earlystop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', min_delta=0.001, patience=3)
resnet_tensorboard = tf.keras.callbacks.TensorBoard(log_dir=f'{directory}/resnet_tensorboard')
resnet_callbacks=[resnet_modelchkpt, resnet_earlystop, resnet_tensorboard]
resnet_fitted_model = resnet_inc_model.fit(aug_train_ds, validation_data=validation_ds, batch_size=32, epochs=5, callbacks=resnet_callbacks)
```

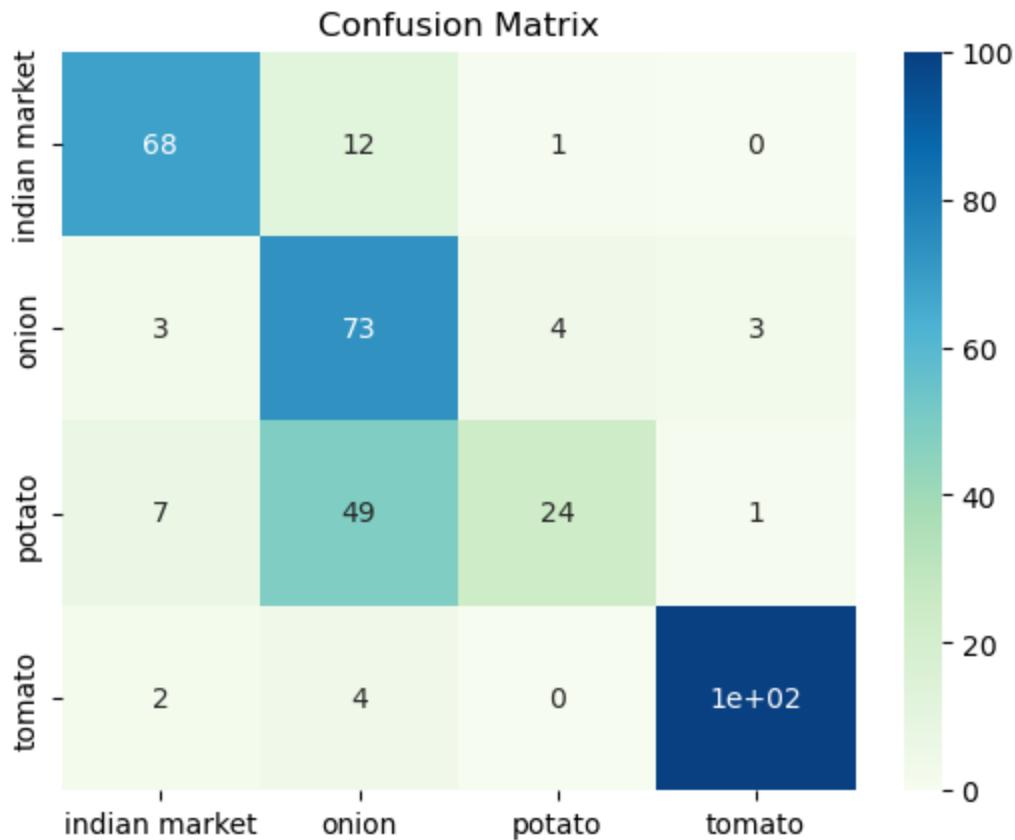
Epoch 1/10  
**79/79** 466s 5s/step - accuracy: 0.6271 - loss: 1.6533  
- val\_accuracy: 0.2935 - val\_loss: 1.9659  
Epoch 2/10  
**79/79** 348s 4s/step - accuracy: 0.7611 - loss: 0.6209  
- val\_accuracy: 0.2935 - val\_loss: 3.5180  
Epoch 3/10  
**79/79** 335s 4s/step - accuracy: 0.8019 - loss: 0.5273  
- val\_accuracy: 0.2648 - val\_loss: 1.5681  
Epoch 4/10  
**79/79** 339s 4s/step - accuracy: 0.8152 - loss: 0.5290  
- val\_accuracy: 0.2791 - val\_loss: 1.3895  
Epoch 5/10  
**79/79** 335s 4s/step - accuracy: 0.8482 - loss: 0.4006  
- val\_accuracy: 0.2791 - val\_loss: 1.4085  
Epoch 6/10  
**79/79** 339s 4s/step - accuracy: 0.8563 - loss: 0.4251  
- val\_accuracy: 0.2663 - val\_loss: 1.5081  
Epoch 7/10

```
79/79 339s 4s/step - accuracy: 0.8537 - loss: 0.4082
- val_accuracy: 0.4051 - val_loss: 1.2852
Epoch 8/10
79/79 338s 4s/step - accuracy: 0.8716 - loss: 0.3816
- val_accuracy: 0.2823 - val_loss: 1.6024
Epoch 9/10
79/79 338s 4s/step - accuracy: 0.8844 - loss: 0.3296
- val_accuracy: 0.5375 - val_loss: 1.2327
Epoch 10/10
79/79 337s 4s/step - accuracy: 0.8753 - loss: 0.3444
- val_accuracy: 0.6826 - val_loss: 1.2252
```

## Testing Accuracy

```
In [67]: resnet_inc_model.load_weights(filepath=f'{directory}/resnet_modelchkpt/best_model.weight')
print_accuracy(resnet_inc_model,test_ds)
print()
print_confusion_matrix(resnet_inc_model,test_ds,show_plot=True)
```

The accuracy of the model on the provided dataset is 75.50%



```
In [68]: built_models['Resnet_model']=resnet_inc_model
```

## Implementation with Inception net

```
In [69]: from tensorflow.keras.applications import InceptionResNetV2
```

```
In [70]: inception_net_model = InceptionResNetV2(include_top=False,weights='imagenet',input_shape
inception_net_model.Trainable=False
inception_net_model.summary())
```

WARNING:tensorflow:From C:\Anaconda\lib\site-packages\keras\src\backend\tensorflow\core.py:184: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

Model: "inception\_resnet\_v2"

Layer (type)	Output Shape	Param #	Connected To
input_layer_9 (InputLayer)	(None, 128, 128, 3)	0	-
conv2d_15 (Conv2D)	(None, 63, 63, 32)	864	input_layer_9
batch_normalization_12 (BatchNormalization)	(None, 63, 63, 32)	96	conv2d_15[0]
activation_14 (Activation)	(None, 63, 63, 32)	0	batch_normalization_12
conv2d_16 (Conv2D)	(None, 61, 61, 32)	9,216	activation_14
batch_normalization_13 (BatchNormalization)	(None, 61, 61, 32)	96	conv2d_16[0]
activation_15 (Activation)	(None, 61, 61, 32)	0	batch_normalization_13
conv2d_17 (Conv2D)	(None, 61, 61, 64)	18,432	activation_15
batch_normalization_14 (BatchNormalization)	(None, 61, 61, 64)	192	conv2d_17[0]
activation_16 (Activation)	(None, 61, 61, 64)	0	batch_normalization_14
max_pooling2d_15 (MaxPooling2D)	(None, 30, 30, 64)	0	activation_16
conv2d_18 (Conv2D)	(None, 30, 30, 80)	5,120	max_pooling2d_15
batch_normalization_15 (BatchNormalization)	(None, 30, 30, 80)	240	conv2d_18[0]
activation_17 (Activation)	(None, 30, 30, 80)	0	batch_normalization_15
conv2d_19 (Conv2D)	(None, 28, 28, 192)	138,240	activation_17
batch_normalization_16 (BatchNormalization)	(None, 28, 28, 192)	576	conv2d_19[0]
activation_18 (Activation)	(None, 28, 28, 192)	0	batch_normalization_16
max_pooling2d_16 (MaxPooling2D)	(None, 13, 13, 192)	0	activation_18
conv2d_23 (Conv2D)	(None, 13, 13, 64)	12,288	max_pooling2d_16
batch_normalization_20 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_23[0]
activation_22 (Activation)	(None, 13, 13, 64)	0	batch_normalization_20
conv2d_21 (Conv2D)	(None, 13, 13, 48)	9,216	max_pooling2d_16
conv2d_24 (Conv2D)	(None, 13, 13, 96)	55,296	activation_22
batch_normalization_18 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_21[0]
batch_normalization_21	(None, 13, 13, 96)	288	conv2d_24[0]

(BatchNormalization)			
activation_20 (Activation)	(None, 13, 13, 48)	0	batch_norm
activation_23 (Activation)	(None, 13, 13, 96)	0	batch_norm
average_pooling2d (AveragePooling2D)	(None, 13, 13, 192)	0	max_poolin
conv2d_20 (Conv2D)	(None, 13, 13, 96)	18,432	max_poolin
conv2d_22 (Conv2D)	(None, 13, 13, 64)	76,800	activation
conv2d_25 (Conv2D)	(None, 13, 13, 96)	82,944	activation
conv2d_26 (Conv2D)	(None, 13, 13, 64)	12,288	average_po
batch_normalization_17 (BatchNormalization)	(None, 13, 13, 96)	288	conv2d_20[
batch_normalization_19 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_22[
batch_normalization_22 (BatchNormalization)	(None, 13, 13, 96)	288	conv2d_25[
batch_normalization_23 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_26[
activation_19 (Activation)	(None, 13, 13, 96)	0	batch_norm
activation_21 (Activation)	(None, 13, 13, 64)	0	batch_norm
activation_24 (Activation)	(None, 13, 13, 96)	0	batch_norm
activation_25 (Activation)	(None, 13, 13, 64)	0	batch_norm
mixed_5b (Concatenate)	(None, 13, 13, 320)	0	activation activation activation activation
conv2d_30 (Conv2D)	(None, 13, 13, 32)	10,240	mixed_5b[0
batch_normalization_27 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_30[
activation_29 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_28 (Conv2D)	(None, 13, 13, 32)	10,240	mixed_5b[0
conv2d_31 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_25 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_28[
batch_normalization_28 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_31[
activation_27 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_30 (Activation)	(None, 13, 13, 48)	0	batch_norm

conv2d_27 (Conv2D)	(None, 13, 13, 32)	10,240	mixed_5b[0]
conv2d_29 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_32 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_24 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_27[0]
batch_normalization_26 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_29[0]
batch_normalization_29 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_32[0]
activation_26 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_28 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_31 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_1_mixed (Concatenate)	(None, 13, 13, 128)	0	activation
activation		activation	activation
block35_1_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_1[0]
custom_scale_layer (CustomScaleLayer)	(None, 13, 13, 320)	0	mixed_5b[0]
block35_1_ac (Activation)	(None, 13, 13, 320)	0	block35_1[0]
conv2d_36 (Conv2D)	(None, 13, 13, 32)	10,240	block35_1[1]
batch_normalization_33 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_36[0]
activation_35 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_34 (Conv2D)	(None, 13, 13, 32)	10,240	block35_1[1]
conv2d_37 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_31 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_34[0]
batch_normalization_34 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_37[0]
activation_33 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_36 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_33 (Conv2D)	(None, 13, 13, 32)	10,240	block35_1[1]
conv2d_35 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_38 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_30 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_33[0]

batch_normalization_32 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_35[
batch_normalization_35 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_38[
activation_32 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_34 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_37 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_2_mixed (Concatenate)	(None, 13, 13, 128)	0	activation activation activation
block35_2_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_2_i
custom_scale_layer_1 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_1_i block35_2_i
block35_2_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_42 (Conv2D)	(None, 13, 13, 32)	10,240	block35_2_i
batch_normalization_39 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_42[
activation_41 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_40 (Conv2D)	(None, 13, 13, 32)	10,240	block35_2_i
conv2d_43 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_37 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_40[
batch_normalization_40 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_43[
activation_39 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_42 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_39 (Conv2D)	(None, 13, 13, 32)	10,240	block35_2_i
conv2d_41 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_44 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_36 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_39[
batch_normalization_38 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_41[
batch_normalization_41 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_44[
activation_38 (Activation)	(None, 13, 13, 32)	0	batch_norm

activation_40 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_43 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_3_mixed (Concatenate)	(None, 13, 13, 128)	0	activation activation activation
block35_3_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_3_i
custom_scale_layer_2 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_2_i block35_3_i
block35_3_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_48 (Conv2D)	(None, 13, 13, 32)	10,240	block35_3_i
batch_normalization_45 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_48[1]
activation_47 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_46 (Conv2D)	(None, 13, 13, 32)	10,240	block35_3_i
conv2d_49 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_43 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_46[1]
batch_normalization_46 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_49[1]
activation_45 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_48 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_45 (Conv2D)	(None, 13, 13, 32)	10,240	block35_3_i
conv2d_47 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_50 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_42 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_45[1]
batch_normalization_44 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_47[1]
batch_normalization_47 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_50[1]
activation_44 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_46 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_49 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_4_mixed (Concatenate)	(None, 13, 13, 128)	0	activation activation activation
block35_4_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_4_i

custom_scale_layer_3 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_3 block35_4
block35_4_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_54 (Conv2D)	(None, 13, 13, 32)	10,240	block35_4_
batch_normalization_51 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_54[
activation_53 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_52 (Conv2D)	(None, 13, 13, 32)	10,240	block35_4_
conv2d_55 (Conv2D)	(None, 13, 13, 48)	13,824	activation_
batch_normalization_49 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_52[
batch_normalization_52 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_55[
activation_51 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_54 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_51 (Conv2D)	(None, 13, 13, 32)	10,240	block35_4_
conv2d_53 (Conv2D)	(None, 13, 13, 32)	9,216	activation_
conv2d_56 (Conv2D)	(None, 13, 13, 64)	27,648	activation_
batch_normalization_48 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_51[
batch_normalization_50 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_53[
batch_normalization_53 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_56[
activation_50 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_52 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_55 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_5_mixed (Concatenate)	(None, 13, 13, 128)	0	activation_
block35_5_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_5_
custom_scale_layer_4 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_4_
block35_5_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_60 (Conv2D)	(None, 13, 13, 32)	10,240	block35_5_
batch_normalization_57	(None, 13, 13, 32)	96	conv2d_60[

(BatchNormalization)			
activation_59 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_58 (Conv2D)	(None, 13, 13, 32)	10,240	block35_5_
conv2d_61 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_55 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_58[
batch_normalization_58 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_61[
activation_57 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_60 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_57 (Conv2D)	(None, 13, 13, 32)	10,240	block35_5_
conv2d_59 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_62 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_54 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_57[
batch_normalization_56 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_59[
batch_normalization_59 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_62[
activation_56 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_58 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_61 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_6_mixed (Concatenate)	(None, 13, 13, 128)	0	activation activation activation
block35_6_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_6_
custom_scale_layer_5 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_5_
block35_6_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_66 (Conv2D)	(None, 13, 13, 32)	10,240	block35_6_
batch_normalization_63 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_66[
activation_65 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_64 (Conv2D)	(None, 13, 13, 32)	10,240	block35_6_
conv2d_67 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_61	(None, 13, 13, 32)	96	conv2d_64[

(BatchNormalization)			
batch_normalization_64 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_67[
activation_63 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_66 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_63 (Conv2D)	(None, 13, 13, 32)	10,240	block35_6
conv2d_65 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_68 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_60 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_63[
batch_normalization_62 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_65[
batch_normalization_65 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_68[
activation_62 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_64 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_67 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_7_mixed (Concatenate)	(None, 13, 13, 128)	0	activation
block35_7_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_7
custom_scale_layer_6 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_6
block35_7_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_72 (Conv2D)	(None, 13, 13, 32)	10,240	block35_7
batch_normalization_69 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_72[
activation_71 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_70 (Conv2D)	(None, 13, 13, 32)	10,240	block35_7
conv2d_73 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_67 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_70[
batch_normalization_70 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_73[
activation_69 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_72 (Activation)	(None, 13, 13, 48)	0	batch_norm

conv2d_69 (Conv2D)	(None, 13, 13, 32)	10,240	block35_7[
conv2d_71 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_74 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_66 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_69[
batch_normalization_68 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_71[
batch_normalization_71 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_74[
activation_68 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_70 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_73 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_8_mixed (Concatenate)	(None, 13, 13, 128)	0	activation
activation		activation	activation
block35_8_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_8[
custom_scale_layer_7 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_7[
block35_8_ac (Activation)	(None, 13, 13, 320)	0	block35_8[
conv2d_78 (Conv2D)	(None, 13, 13, 32)	10,240	block35_8[
batch_normalization_75 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_78[
activation_77 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_76 (Conv2D)	(None, 13, 13, 32)	10,240	block35_8[
conv2d_79 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_73 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_76[
batch_normalization_76 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_79[
activation_75 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_78 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_75 (Conv2D)	(None, 13, 13, 32)	10,240	block35_8[
conv2d_77 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_80 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_72 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_75[

batch_normalization_74 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_77[
batch_normalization_77 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_80[
activation_74 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_76 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_79 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_9_mixed (Concatenate)	(None, 13, 13, 128)	0	activation activation activation
block35_9_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_9[
custom_scale_layer_8 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_8[ block35_9[
block35_9_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_84 (Conv2D)	(None, 13, 13, 32)	10,240	block35_9[
batch_normalization_81 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_84[
activation_83 (Activation)	(None, 13, 13, 32)	0	batch_norm
conv2d_82 (Conv2D)	(None, 13, 13, 32)	10,240	block35_9[
conv2d_85 (Conv2D)	(None, 13, 13, 48)	13,824	activation
batch_normalization_79 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_82[
batch_normalization_82 (BatchNormalization)	(None, 13, 13, 48)	144	conv2d_85[
activation_81 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_84 (Activation)	(None, 13, 13, 48)	0	batch_norm
conv2d_81 (Conv2D)	(None, 13, 13, 32)	10,240	block35_9[
conv2d_83 (Conv2D)	(None, 13, 13, 32)	9,216	activation
conv2d_86 (Conv2D)	(None, 13, 13, 64)	27,648	activation
batch_normalization_78 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_81[
batch_normalization_80 (BatchNormalization)	(None, 13, 13, 32)	96	conv2d_83[
batch_normalization_83 (BatchNormalization)	(None, 13, 13, 64)	192	conv2d_86[
activation_80 (Activation)	(None, 13, 13, 32)	0	batch_norm
activation_82 (Activation)	(None, 13, 13, 32)	0	batch_norm

activation_85 (Activation)	(None, 13, 13, 64)	0	batch_norm
block35_10_mixed (Concatenate)	(None, 13, 13, 128)	0	activation_activation_activation
block35_10_conv (Conv2D)	(None, 13, 13, 320)	41,280	block35_10
custom_scale_layer_9 (CustomScaleLayer)	(None, 13, 13, 320)	0	block35_9_block35_10
block35_10_ac (Activation)	(None, 13, 13, 320)	0	custom_sca
conv2d_88 (Conv2D)	(None, 13, 13, 256)	81,920	block35_10
batch_normalization_85 (BatchNormalization)	(None, 13, 13, 256)	768	conv2d_88[1]
activation_87 (Activation)	(None, 13, 13, 256)	0	batch_norm
conv2d_89 (Conv2D)	(None, 13, 13, 256)	589,824	activation
batch_normalization_86 (BatchNormalization)	(None, 13, 13, 256)	768	conv2d_89[1]
activation_88 (Activation)	(None, 13, 13, 256)	0	batch_norm
conv2d_87 (Conv2D)	(None, 6, 6, 384)	1,105,920	block35_10
conv2d_90 (Conv2D)	(None, 6, 6, 384)	884,736	activation
batch_normalization_84 (BatchNormalization)	(None, 6, 6, 384)	1,152	conv2d_87[1]
batch_normalization_87 (BatchNormalization)	(None, 6, 6, 384)	1,152	conv2d_90[1]
activation_86 (Activation)	(None, 6, 6, 384)	0	batch_norm
activation_89 (Activation)	(None, 6, 6, 384)	0	batch_norm
max_pooling2d_17 (MaxPooling2D)	(None, 6, 6, 320)	0	block35_10
mixed_6a (Concatenate)	(None, 6, 6, 1088)	0	activation_activation_max_pooling
conv2d_92 (Conv2D)	(None, 6, 6, 128)	139,264	mixed_6a[0]
batch_normalization_89 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_92[1]
activation_91 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_93 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_90 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_93[1]
activation_92 (Activation)	(None, 6, 6, 160)	0	batch_norm

conv2d_91 (Conv2D)	(None, 6, 6, 192)	208,896	mixed_6a[0]
conv2d_94 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_88 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_91[1]
batch_normalization_91 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_94[1]
activation_90 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_93 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_1_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_1_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_1_1
custom_scale_layer_10 (CustomScaleLayer)	(None, 6, 6, 1088)	0	mixed_6a[0] block17_1_1
block17_1_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_96 (Conv2D)	(None, 6, 6, 128)	139,264	block17_1_1
batch_normalization_93 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_96[1]
activation_95 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_97 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_94 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_97[1]
activation_96 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_95 (Conv2D)	(None, 6, 6, 192)	208,896	block17_1_2
conv2d_98 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_92 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_95[1]
batch_normalization_95 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_98[1]
activation_94 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_97 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_2_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_2_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_2_1
custom_scale_layer_11 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_1_2 block17_2_1
block17_2_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca

conv2d_100 (Conv2D)	(None, 6, 6, 128)	139,264	block17_2_
batch_normalization_97 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_100
activation_99 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_101 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_98 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_101
activation_100 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_99 (Conv2D)	(None, 6, 6, 192)	208,896	block17_2_
conv2d_102 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_96 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_99[
batch_normalization_99 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_102
activation_98 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_101 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_3_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_3_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_3_
custom_scale_layer_12 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_2_ block17_3_
block17_3_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_104 (Conv2D)	(None, 6, 6, 128)	139,264	block17_3_
batch_normalization_101 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_104
activation_103 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_105 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_102 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_105
activation_104 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_103 (Conv2D)	(None, 6, 6, 192)	208,896	block17_3_
conv2d_106 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_100 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_103
batch_normalization_103 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_106

activation_102 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_105 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_4_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_4_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_4_i
custom_scale_layer_13 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_3_i block17_4_i
block17_4_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_108 (Conv2D)	(None, 6, 6, 128)	139,264	block17_4_i
batch_normalization_105 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_108
activation_107 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_109 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_106 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_109
activation_108 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_107 (Conv2D)	(None, 6, 6, 192)	208,896	block17_4_i
conv2d_110 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_104 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_107
batch_normalization_107 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_110
activation_106 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_109 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_5_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_5_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_5_i
custom_scale_layer_14 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_4_i block17_5_i
block17_5_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_112 (Conv2D)	(None, 6, 6, 128)	139,264	block17_5_i
batch_normalization_109 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_112
activation_111 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_113 (Conv2D)	(None, 6, 6, 160)	143,360	activation

batch_normalization_110 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_113
activation_112 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_111 (Conv2D)	(None, 6, 6, 192)	208,896	block17_5_
conv2d_114 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_108 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_111
batch_normalization_111 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_114
activation_110 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_113 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_6_mixed (Concatenate)	(None, 6, 6, 384)	0	activation
block17_6_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_6_1
custom_scale_layer_15 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_5_1
block17_6_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_116 (Conv2D)	(None, 6, 6, 128)	139,264	block17_6_2
batch_normalization_113 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_116
activation_115 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_117 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_114 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_117
activation_116 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_115 (Conv2D)	(None, 6, 6, 192)	208,896	block17_6_3
conv2d_118 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_112 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_115
batch_normalization_115 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_118
activation_114 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_117 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_7_mixed (Concatenate)	(None, 6, 6, 384)	0	activation
block17_7_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_7_1

custom_scale_layer_16 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_6 block17_7_1
block17_7_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_120 (Conv2D)	(None, 6, 6, 128)	139,264	block17_7_1
batch_normalization_117 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_120
activation_119 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_121 (Conv2D)	(None, 6, 6, 160)	143,360	activation_119
batch_normalization_118 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_121
activation_120 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_119 (Conv2D)	(None, 6, 6, 192)	208,896	block17_7_1
conv2d_122 (Conv2D)	(None, 6, 6, 192)	215,040	activation_120
batch_normalization_116 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_119
batch_normalization_119 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_122
activation_118 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_121 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_8_mixed (Concatenate)	(None, 6, 6, 384)	0	activation_121 activation_122
block17_8_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_8_1
custom_scale_layer_17 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_7_1 block17_8_1
block17_8_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_124 (Conv2D)	(None, 6, 6, 128)	139,264	block17_8_1
batch_normalization_121 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_124
activation_123 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_125 (Conv2D)	(None, 6, 6, 160)	143,360	activation_123
batch_normalization_122 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_125
activation_124 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_123 (Conv2D)	(None, 6, 6, 192)	208,896	block17_8_1
conv2d_126 (Conv2D)	(None, 6, 6, 192)	215,040	activation_124
batch_normalization_120	(None, 6, 6, 192)	576	conv2d_123

(BatchNormalization)			
batch_normalization_123 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_126
activation_122 (Activation)	(None, 6, 6, 192)	0	batch_norm_1
activation_125 (Activation)	(None, 6, 6, 192)	0	batch_norm_1
block17_9_mixed (Concatenate)	(None, 6, 6, 384)	0	activation_1
block17_9_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_9_1
custom_scale_layer_18 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_8_1
block17_9_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_128 (Conv2D)	(None, 6, 6, 128)	139,264	block17_9_2
batch_normalization_125 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_128
activation_127 (Activation)	(None, 6, 6, 128)	0	batch_norm_2
conv2d_129 (Conv2D)	(None, 6, 6, 160)	143,360	activation_2
batch_normalization_126 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_129
activation_128 (Activation)	(None, 6, 6, 160)	0	batch_norm_2
conv2d_127 (Conv2D)	(None, 6, 6, 192)	208,896	block17_9_3
conv2d_130 (Conv2D)	(None, 6, 6, 192)	215,040	activation_3
batch_normalization_124 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_127
batch_normalization_127 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_130
activation_126 (Activation)	(None, 6, 6, 192)	0	batch_norm_3
activation_129 (Activation)	(None, 6, 6, 192)	0	batch_norm_3
block17_10_mixed (Concatenate)	(None, 6, 6, 384)	0	activation_3
block17_10_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_10_1
custom_scale_layer_19 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_9_2
block17_10_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_132 (Conv2D)	(None, 6, 6, 128)	139,264	block17_10_2
batch_normalization_129 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_132

activation_131 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_133 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_130 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_133
activation_132 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_131 (Conv2D)	(None, 6, 6, 192)	208,896	block17_10
conv2d_134 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_128 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_131
batch_normalization_131 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_134
activation_130 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_133 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_11_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_11_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_11
custom_scale_layer_20 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_10 block17_11
block17_11_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_136 (Conv2D)	(None, 6, 6, 128)	139,264	block17_11
batch_normalization_133 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_136
activation_135 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_137 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_134 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_137
activation_136 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_135 (Conv2D)	(None, 6, 6, 192)	208,896	block17_11
conv2d_138 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_132 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_135
batch_normalization_135 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_138
activation_134 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_137 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_12_mixed	(None, 6, 6, 384)	0	activation

(Concatenate)				activation
block17_12_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_12	
custom_scale_layer_21 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_11	block17_12
block17_12_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca	
conv2d_140 (Conv2D)	(None, 6, 6, 128)	139,264	block17_12	
batch_normalization_137 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_140	
activation_139 (Activation)	(None, 6, 6, 128)	0	batch_norm	
conv2d_141 (Conv2D)	(None, 6, 6, 160)	143,360	activation	
batch_normalization_138 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_141	
activation_140 (Activation)	(None, 6, 6, 160)	0	batch_norm	
conv2d_139 (Conv2D)	(None, 6, 6, 192)	208,896	block17_12	
conv2d_142 (Conv2D)	(None, 6, 6, 192)	215,040	activation	
batch_normalization_136 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_139	
batch_normalization_139 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_142	
activation_138 (Activation)	(None, 6, 6, 192)	0	batch_norm	
activation_141 (Activation)	(None, 6, 6, 192)	0	batch_norm	
block17_13_mixed (Concatenate)	(None, 6, 6, 384)	0	activation	activation
block17_13_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_13	
custom_scale_layer_22 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_12	block17_13
block17_13_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca	
conv2d_144 (Conv2D)	(None, 6, 6, 128)	139,264	block17_13	
batch_normalization_141 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_144	
activation_143 (Activation)	(None, 6, 6, 128)	0	batch_norm	
conv2d_145 (Conv2D)	(None, 6, 6, 160)	143,360	activation	
batch_normalization_142 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_145	
activation_144 (Activation)	(None, 6, 6, 160)	0	batch_norm	
conv2d_143 (Conv2D)	(None, 6, 6, 192)	208,896	block17_13	

conv2d_146 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_140 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_143
batch_normalization_143 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_146
activation_142 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_145 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_14_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_14_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_14
custom_scale_layer_23 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_13 block17_14
block17_14_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_148 (Conv2D)	(None, 6, 6, 128)	139,264	block17_14
batch_normalization_145 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_148
activation_147 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_149 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_146 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_149
activation_148 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_147 (Conv2D)	(None, 6, 6, 192)	208,896	block17_14
conv2d_150 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_144 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_147
batch_normalization_147 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_150
activation_146 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_149 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_15_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_15_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_15
custom_scale_layer_24 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_14 block17_15
block17_15_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_152 (Conv2D)	(None, 6, 6, 128)	139,264	block17_15

batch_normalization_149 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_152
activation_151 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_153 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_150 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_153
activation_152 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_151 (Conv2D)	(None, 6, 6, 192)	208,896	block17_15
conv2d_154 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_148 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_151
batch_normalization_151 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_154
activation_150 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_153 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_16_mixed (Concatenate)	(None, 6, 6, 384)	0	activation
block17_16_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_16
custom_scale_layer_25 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_15
block17_16_ac (Activation)	(None, 6, 6, 1088)	0	block17_16
conv2d_156 (Conv2D)	(None, 6, 6, 128)	139,264	block17_16
batch_normalization_153 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_156
activation_155 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_157 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_154 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_157
activation_156 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_155 (Conv2D)	(None, 6, 6, 192)	208,896	block17_16
conv2d_158 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_152 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_155
batch_normalization_155 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_158
activation_154 (Activation)	(None, 6, 6, 192)	0	batch_norm

activation_157 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_17_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_17_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_17
custom_scale_layer_26 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_16 block17_17
block17_17_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_160 (Conv2D)	(None, 6, 6, 128)	139,264	block17_17
batch_normalization_157 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_160
activation_159 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_161 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_158 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_161
activation_160 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_159 (Conv2D)	(None, 6, 6, 192)	208,896	block17_17
conv2d_162 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_156 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_159
batch_normalization_159 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_162
activation_158 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_161 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_18_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_18_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_18
custom_scale_layer_27 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_17 block17_18
block17_18_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_164 (Conv2D)	(None, 6, 6, 128)	139,264	block17_18
batch_normalization_161 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_164
activation_163 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_165 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_162 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_165

activation_164 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_163 (Conv2D)	(None, 6, 6, 192)	208,896	block17_18
conv2d_166 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_160 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_163
batch_normalization_163 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_166
activation_162 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_165 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_19_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_19_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_19
custom_scale_layer_28 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_18 block17_19
block17_19_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_168 (Conv2D)	(None, 6, 6, 128)	139,264	block17_19
batch_normalization_165 (BatchNormalization)	(None, 6, 6, 128)	384	conv2d_168
activation_167 (Activation)	(None, 6, 6, 128)	0	batch_norm
conv2d_169 (Conv2D)	(None, 6, 6, 160)	143,360	activation
batch_normalization_166 (BatchNormalization)	(None, 6, 6, 160)	480	conv2d_169
activation_168 (Activation)	(None, 6, 6, 160)	0	batch_norm
conv2d_167 (Conv2D)	(None, 6, 6, 192)	208,896	block17_19
conv2d_170 (Conv2D)	(None, 6, 6, 192)	215,040	activation
batch_normalization_164 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_167
batch_normalization_167 (BatchNormalization)	(None, 6, 6, 192)	576	conv2d_170
activation_166 (Activation)	(None, 6, 6, 192)	0	batch_norm
activation_169 (Activation)	(None, 6, 6, 192)	0	batch_norm
block17_20_mixed (Concatenate)	(None, 6, 6, 384)	0	activation activation
block17_20_conv (Conv2D)	(None, 6, 6, 1088)	418,880	block17_20
custom_scale_layer_29 (CustomScaleLayer)	(None, 6, 6, 1088)	0	block17_19 block17_20

block17_20_ac (Activation)	(None, 6, 6, 1088)	0	custom_sca
conv2d_175 (Conv2D)	(None, 6, 6, 256)	278,528	block17_20
batch_normalization_172 (BatchNormalization)	(None, 6, 6, 256)	768	conv2d_175
activation_174 (Activation)	(None, 6, 6, 256)	0	batch_norm
conv2d_171 (Conv2D)	(None, 6, 6, 256)	278,528	block17_20
conv2d_173 (Conv2D)	(None, 6, 6, 256)	278,528	block17_20
conv2d_176 (Conv2D)	(None, 6, 6, 288)	663,552	activation
batch_normalization_168 (BatchNormalization)	(None, 6, 6, 256)	768	conv2d_171
batch_normalization_170 (BatchNormalization)	(None, 6, 6, 256)	768	conv2d_173
batch_normalization_173 (BatchNormalization)	(None, 6, 6, 288)	864	conv2d_176
activation_170 (Activation)	(None, 6, 6, 256)	0	batch_norm
activation_172 (Activation)	(None, 6, 6, 256)	0	batch_norm
activation_175 (Activation)	(None, 6, 6, 288)	0	batch_norm
conv2d_172 (Conv2D)	(None, 2, 2, 384)	884,736	activation
conv2d_174 (Conv2D)	(None, 2, 2, 288)	663,552	activation
conv2d_177 (Conv2D)	(None, 2, 2, 320)	829,440	activation
batch_normalization_169 (BatchNormalization)	(None, 2, 2, 384)	1,152	conv2d_172
batch_normalization_171 (BatchNormalization)	(None, 2, 2, 288)	864	conv2d_174
batch_normalization_174 (BatchNormalization)	(None, 2, 2, 320)	960	conv2d_177
activation_171 (Activation)	(None, 2, 2, 384)	0	batch_norm
activation_173 (Activation)	(None, 2, 2, 288)	0	batch_norm
activation_176 (Activation)	(None, 2, 2, 320)	0	batch_norm
max_pooling2d_18 (MaxPooling2D)	(None, 2, 2, 1088)	0	block17_20
mixed_7a (Concatenate)	(None, 2, 2, 2080)	0	activation activation activation max_pooling
conv2d_179 (Conv2D)	(None, 2, 2, 192)	399,360	mixed_7a[0]

batch_normalization_176 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_179
activation_178 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_180 (Conv2D)	(None, 2, 2, 224)	129,024	activation
batch_normalization_177 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_180
activation_179 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_178 (Conv2D)	(None, 2, 2, 192)	399,360	mixed_7a[0]
conv2d_181 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_175 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_178
batch_normalization_178 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_181
activation_177 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_180 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_1_mixed (Concatenate)	(None, 2, 2, 448)	0	activation
block8_1_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_1_m
custom_scale_layer_30 (CustomScaleLayer)	(None, 2, 2, 2080)	0	mixed_7a[0]
block8_1_ac (Activation)	(None, 2, 2, 2080)	0	custom_sca
conv2d_183 (Conv2D)	(None, 2, 2, 192)	399,360	block8_1_a
batch_normalization_180 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_183
activation_182 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_184 (Conv2D)	(None, 2, 2, 224)	129,024	activation
batch_normalization_181 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_184
activation_183 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_182 (Conv2D)	(None, 2, 2, 192)	399,360	block8_1_a
conv2d_185 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_179 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_182
batch_normalization_182 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_185
activation_181 (Activation)	(None, 2, 2, 192)	0	batch_norm

activation_184 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_2_mixed (Concatenate)	(None, 2, 2, 448)	0	activation_activation
block8_2_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_2_m
custom_scale_layer_31 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_1_a block8_2_c
block8_2_ac (Activation)	(None, 2, 2, 2080)	0	custom_sca
conv2d_187 (Conv2D)	(None, 2, 2, 192)	399,360	block8_2_a
batch_normalization_184 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_187
activation_186 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_188 (Conv2D)	(None, 2, 2, 224)	129,024	activation
batch_normalization_185 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_188
activation_187 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_186 (Conv2D)	(None, 2, 2, 192)	399,360	block8_2_a
conv2d_189 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_183 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_186
batch_normalization_186 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_189
activation_185 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_188 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_3_mixed (Concatenate)	(None, 2, 2, 448)	0	activation_activation
block8_3_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_3_m
custom_scale_layer_32 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_2_a block8_3_c
block8_3_ac (Activation)	(None, 2, 2, 2080)	0	custom_sca
conv2d_191 (Conv2D)	(None, 2, 2, 192)	399,360	block8_3_a
batch_normalization_188 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_191
activation_190 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_192 (Conv2D)	(None, 2, 2, 224)	129,024	activation
batch_normalization_189 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_192

activation_191 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_190 (Conv2D)	(None, 2, 2, 192)	399,360	block8_3_a
conv2d_193 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_187 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_190
batch_normalization_190 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_193
activation_189 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_192 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_4_mixed (Concatenate)	(None, 2, 2, 448)	0	activation
block8_4_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_4_m
custom_scale_layer_33 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_3_a
block8_4_ac (Activation)	(None, 2, 2, 2080)	0	block8_4_c
conv2d_195 (Conv2D)	(None, 2, 2, 192)	399,360	block8_4_a
batch_normalization_192 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_195
activation_194 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_196 (Conv2D)	(None, 2, 2, 224)	129,024	activation
batch_normalization_193 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_196
activation_195 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_194 (Conv2D)	(None, 2, 2, 192)	399,360	block8_4_a
conv2d_197 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_191 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_194
batch_normalization_194 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_197
activation_193 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_196 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_5_mixed (Concatenate)	(None, 2, 2, 448)	0	activation
block8_5_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_5_m
custom_scale_layer_34 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_4_a
			block8_5_c

block8_5_ac (Activation)	(None, 2, 2, 2080)	0	custom_scale_layer_35 (CustomScaleLayer)
conv2d_199 (Conv2D)	(None, 2, 2, 192)	399,360	block8_5_ac (Activation)
batch_normalization_196 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_199 (Conv2D)
activation_198 (Activation)	(None, 2, 2, 192)	0	batch_normalization_197 (BatchNormalization)
conv2d_200 (Conv2D)	(None, 2, 2, 224)	129,024	activation_198 (Activation)
batch_normalization_197 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_200 (Conv2D)
activation_199 (Activation)	(None, 2, 2, 224)	0	batch_normalization_195 (BatchNormalization)
conv2d_198 (Conv2D)	(None, 2, 2, 192)	399,360	block8_5_ac (Activation)
conv2d_201 (Conv2D)	(None, 2, 2, 256)	172,032	activation_197 (Activation)
batch_normalization_195 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_198 (Conv2D)
batch_normalization_198 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_201 (Conv2D)
activation_197 (Activation)	(None, 2, 2, 192)	0	batch_normalization_198 (BatchNormalization)
activation_200 (Activation)	(None, 2, 2, 256)	0	activation_197 (Activation)
block8_6_mixed (Concatenate)	(None, 2, 2, 448)	0	activation_200 (Activation)
block8_6_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_6_mixed (Concatenate)
custom_scale_layer_35 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_5_ac (Activation)
block8_6_ac (Activation)	(None, 2, 2, 2080)	0	custom_scale_layer_35 (CustomScaleLayer)
conv2d_203 (Conv2D)	(None, 2, 2, 192)	399,360	block8_6_ac (Activation)
batch_normalization_200 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_203 (Conv2D)
activation_202 (Activation)	(None, 2, 2, 192)	0	batch_normalization_200 (BatchNormalization)
conv2d_204 (Conv2D)	(None, 2, 2, 224)	129,024	activation_202 (Activation)
batch_normalization_201 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_204 (Conv2D)
activation_203 (Activation)	(None, 2, 2, 224)	0	batch_normalization_201 (BatchNormalization)
conv2d_202 (Conv2D)	(None, 2, 2, 192)	399,360	block8_6_ac (Activation)
conv2d_205 (Conv2D)	(None, 2, 2, 256)	172,032	activation_203 (Activation)
batch_normalization_199 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_202 (Conv2D)
batch_normalization_202 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_205 (Conv2D)

(BatchNormalization)			
activation_201 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_204 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_7_mixed (Concatenate)	(None, 2, 2, 448)	0	activation activation
block8_7_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_7_m
custom_scale_layer_36 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_6_a block8_7_c
block8_7_ac (Activation)	(None, 2, 2, 2080)	0	custom_sca
conv2d_207 (Conv2D)	(None, 2, 2, 192)	399,360	block8_7_a
batch_normalization_204 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_207
activation_206 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_208 (Conv2D)	(None, 2, 2, 224)	129,024	activation
batch_normalization_205 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_208
activation_207 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_206 (Conv2D)	(None, 2, 2, 192)	399,360	block8_7_a
conv2d_209 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_203 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_206
batch_normalization_206 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_209
activation_205 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_208 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_8_mixed (Concatenate)	(None, 2, 2, 448)	0	activation activation
block8_8_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_8_m
custom_scale_layer_37 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_7_a block8_8_c
block8_8_ac (Activation)	(None, 2, 2, 2080)	0	custom_sca
conv2d_211 (Conv2D)	(None, 2, 2, 192)	399,360	block8_8_a
batch_normalization_208 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_211
activation_210 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_212 (Conv2D)	(None, 2, 2, 224)	129,024	activation

batch_normalization_209 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_212
activation_211 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_210 (Conv2D)	(None, 2, 2, 192)	399,360	block8_8_a
conv2d_213 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_207 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_210
batch_normalization_210 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_213
activation_209 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_212 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_9_mixed (Concatenate)	(None, 2, 2, 448)	0	activation
block8_9_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_9_m
custom_scale_layer_38 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_8_a
block8_9_ac (Activation)	(None, 2, 2, 2080)	0	custom_sca
conv2d_215 (Conv2D)	(None, 2, 2, 192)	399,360	block8_9_a
batch_normalization_212 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_215
activation_214 (Activation)	(None, 2, 2, 192)	0	batch_norm
conv2d_216 (Conv2D)	(None, 2, 2, 224)	129,024	activation
batch_normalization_213 (BatchNormalization)	(None, 2, 2, 224)	672	conv2d_216
activation_215 (Activation)	(None, 2, 2, 224)	0	batch_norm
conv2d_214 (Conv2D)	(None, 2, 2, 192)	399,360	block8_9_a
conv2d_217 (Conv2D)	(None, 2, 2, 256)	172,032	activation
batch_normalization_211 (BatchNormalization)	(None, 2, 2, 192)	576	conv2d_214
batch_normalization_214 (BatchNormalization)	(None, 2, 2, 256)	768	conv2d_217
activation_213 (Activation)	(None, 2, 2, 192)	0	batch_norm
activation_216 (Activation)	(None, 2, 2, 256)	0	batch_norm
block8_10_mixed (Concatenate)	(None, 2, 2, 448)	0	activation
block8_10_conv (Conv2D)	(None, 2, 2, 2080)	933,920	block8_10_i

custom_scale_layer_39 (CustomScaleLayer)	(None, 2, 2, 2080)	0	block8_9_a block8_10_a
conv_7b (Conv2D)	(None, 2, 2, 1536)	3,194,880	custom_sca
conv_7b_bn (BatchNormalization)	(None, 2, 2, 1536)	4,608	conv_7b[0]
conv_7b_ac (Activation)	(None, 2, 2, 1536)	0	conv_7b_bn
global_average_pooling2d_4 (GlobalAveragePooling2D)	(None, 1536)	0	conv_7b_ac

Total params: 54,336,736 (207.28 MB)

Trainable params: 54,276,192 (207.05 MB)

Non-trainable params: 60,544 (236.50 KB)

Creating the Sequential model

```
In [71]: inception_inc_model = Sequential(
    name='inception_included_model',
    layers=[
        inception_net_model,
        tf.keras.layers.Dense(units=1536, activation='relu', use_bias=True), #Adding only
        tf.keras.layers.Dense(units=4, activation='softmax', use_bias=True)
    ]
)

inception_inc_model.summary()
```

Model: "inception\_included\_model"

Layer (type)	Output Shape	Param #
inception_resnet_v2 (Functional)	?	54,336,736
dense_11 (Dense)	?	0 (unbuilt)
dense_12 (Dense)	?	0 (unbuilt)

Total params: 54,336,736 (207.28 MB)

Trainable params: 54,276,192 (207.05 MB)

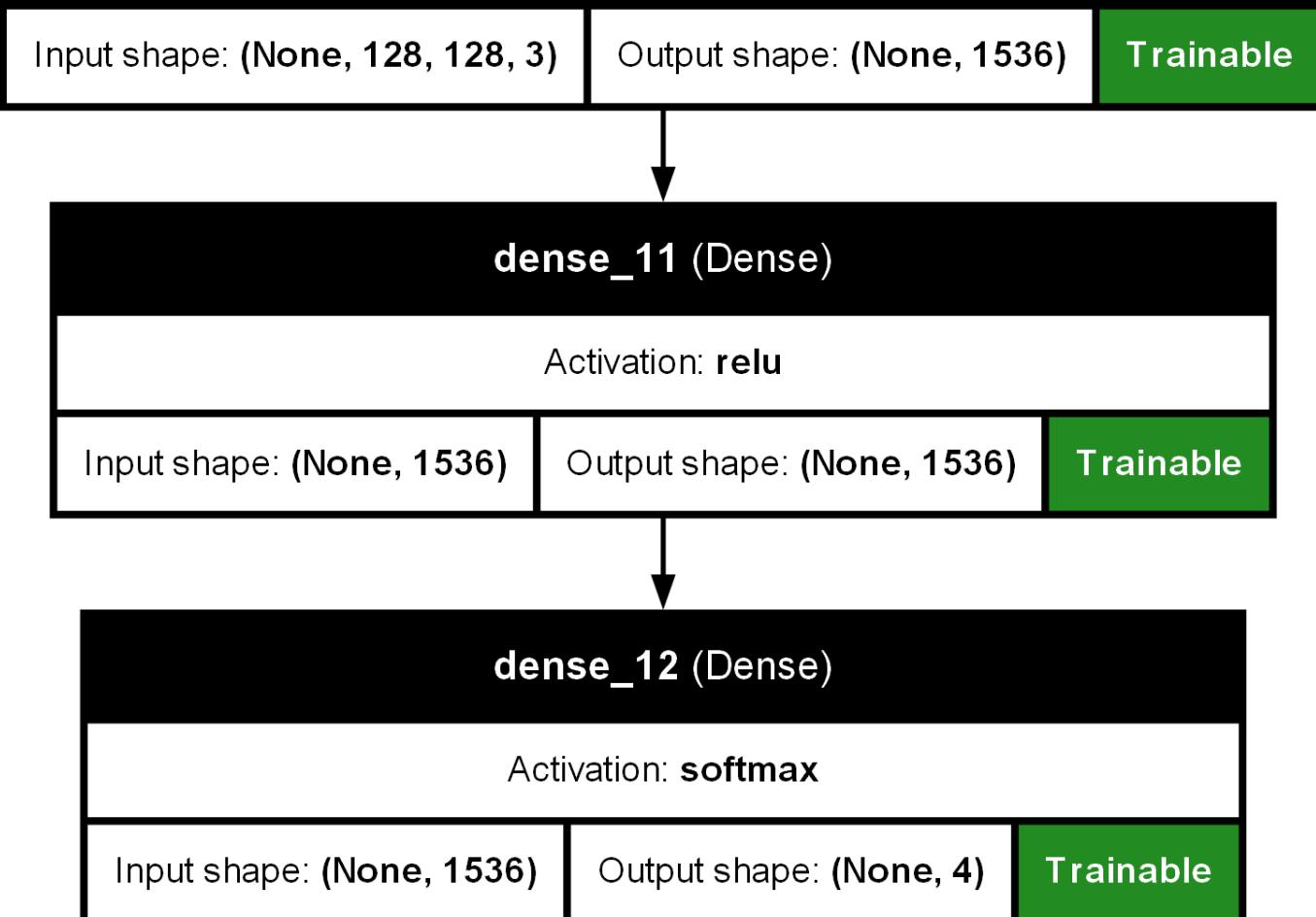
Non-trainable params: 60,544 (236.50 KB)

Plotting the model

```
In [153... tf.keras.utils.plot_model(inception_inc_model, show_layer_names=True, show_trainable=True,
```

Out[153]:

## inception\_resnet\_v2 (Functional)



### Inception Model Compilation

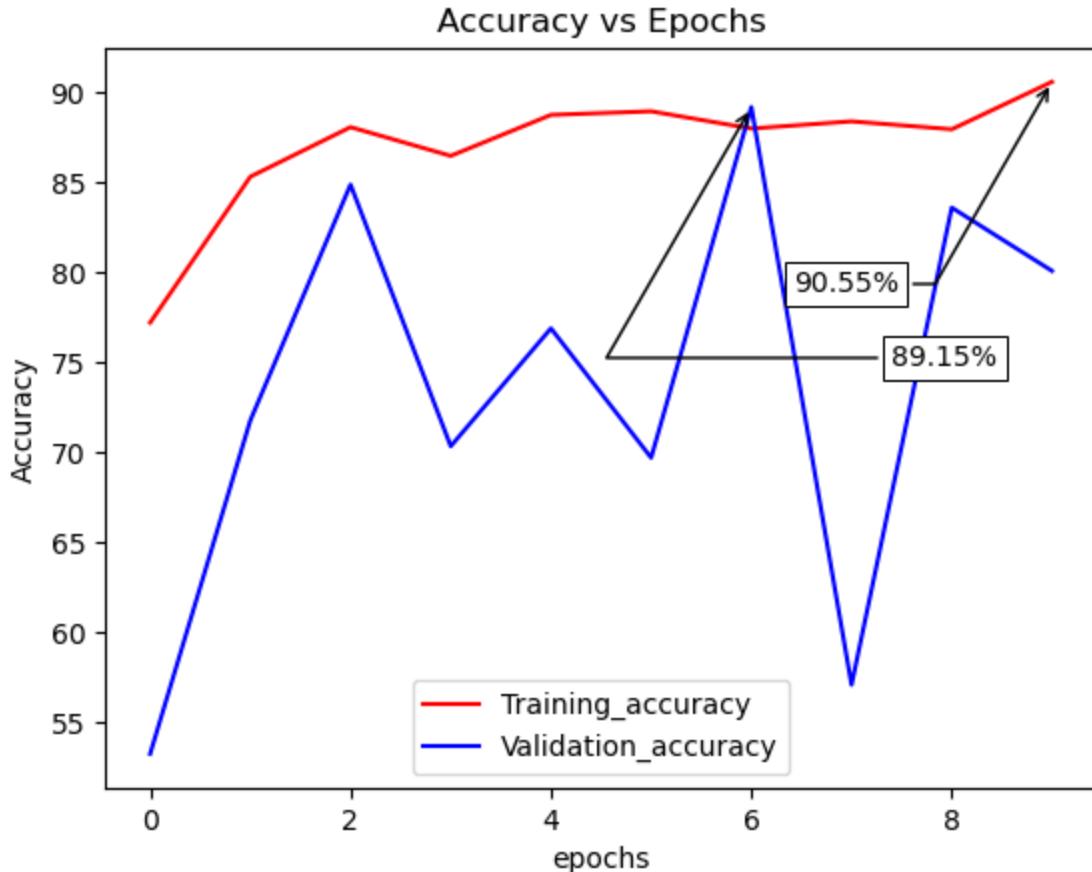
```
In [72]: inception_inc_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
inception_directory = './inception_model'
inception_checkpoint = tf.keras.callbacks.ModelCheckpoint(filepath=f'{inception_directory}/inception.h5')
inception_tensorboard = tf.keras.callbacks.TensorBoard(log_dir=f'{inception_directory}/inception_tboard')
inception_earlystop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10, mode='min')
inception_callbacks = [inception_checkpoint, inception_tensorboard, inception_earlystop]
fitted_inception_model = inception_inc_model.fit(aug_train_ds, validation_data=validation_ds, epochs=100)

Epoch 1/10
79/79 660s 5s/step - accuracy: 0.6783 - loss: 1.0869
- val_accuracy: 0.5327 - val_loss: 3.3232
Epoch 2/10
79/79 420s 5s/step - accuracy: 0.8482 - loss: 0.4207
- val_accuracy: 0.7177 - val_loss: 1.0044
Epoch 3/10
79/79 441s 6s/step - accuracy: 0.8818 - loss: 0.3480
- val_accuracy: 0.8485 - val_loss: 0.4549
Epoch 4/10
79/79 404s 5s/step - accuracy: 0.8577 - loss: 0.4113
- val_accuracy: 0.7033 - val_loss: 1.6918
Epoch 5/10
79/79 401s 5s/step - accuracy: 0.8912 - loss: 0.2886
- val_accuracy: 0.7687 - val_loss: 7.4390
Epoch 6/10
79/79 393s 5s/step - accuracy: 0.8936 - loss: 0.3019
- val_accuracy: 0.6970 - val_loss: 1.5366
Epoch 7/10
79/79 416s 5s/step - accuracy: 0.8788 - loss: 0.3379
```

```
- val_accuracy: 0.8915 - val_loss: 0.2623
Epoch 8/10
79/79 387s 5s/step - accuracy: 0.8917 - loss: 0.3080
- val_accuracy: 0.5710 - val_loss: 1.6006
Epoch 9/10
79/79 386s 5s/step - accuracy: 0.8802 - loss: 0.3320
- val_accuracy: 0.8357 - val_loss: 0.5564
Epoch 10/10
79/79 394s 5s/step - accuracy: 0.9146 - loss: 0.2426
- val_accuracy: 0.8006 - val_loss: 0.6766
```

Printing Training and Validation accuracy

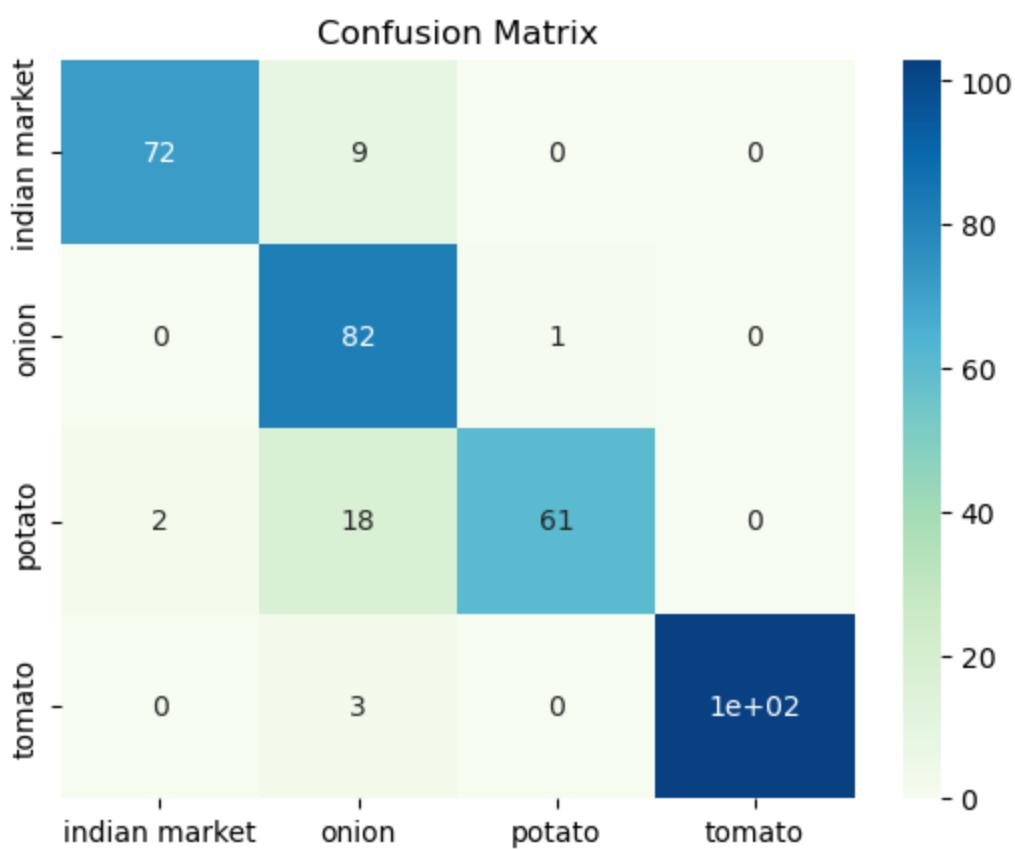
```
In [73]: plot_train_val_accuracy(fitted_incep_model)
```



Testing accuracy and Confusion matrix

```
In [74]: inception_inc_model.load_weights(filepath=f'{incep_directory}/incep_modelchk/incep_best')
print_accuracy(inception_inc_model,test_ds)
print()
print_confusion_matrix(inception_inc_model,test_ds,show_plot=True)
```

The accuracy of the model on the provided dataset is 90.60%



```
In [75]: built_models['Inceptionnet_model']=inception_inc_model
```

## Implementation using MobileNet architecture

```
In [76]: from tensorflow.keras.applications import MobileNetV2
```

```
In [77]: mobilenet_model = MobileNetV2(include_top=False, input_shape=(128, 128, 3), weights='imagenet')
mobilenet_model.Trainable=False
mobilenet_model.summary()
```

Model: "mobilenetv2\_1.00\_128"

Layer (type)	Output Shape	Param #	Connected To
input_layer_11 (InputLayer)	(None, 128, 128, 3)	0	-
Conv1 (Conv2D)	(None, 64, 64, 32)	864	input_layer_11
bn_Conv1 (BatchNormalization)	(None, 64, 64, 32)	128	Conv1[0][0]
Conv1_relu (ReLU)	(None, 64, 64, 32)	0	bn_Conv1[0]
expanded_conv_depthwise (DepthwiseConv2D)	(None, 64, 64, 32)	288	Conv1_relu
expanded_conv_depthwise_BN (BatchNormalization)	(None, 64, 64, 32)	128	expanded_conv_depthwise
expanded_conv_depthwise_relu (ReLU)	(None, 64, 64, 32)	0	expanded_conv_depthwise
expanded_conv_project (Conv2D)	(None, 64, 64, 16)	512	expanded_conv_depthwise

expanded_conv_project_BN (BatchNormalization)	(None, 64, 64, 16)	64	expanded_c
block_1_expand (Conv2D)	(None, 64, 64, 96)	1,536	expanded_c
block_1_expand_BN (BatchNormalization)	(None, 64, 64, 96)	384	block_1_ex
block_1_expand_relu (ReLU)	(None, 64, 64, 96)	0	block_1_ex
block_1_pad (ZeroPadding2D)	(None, 65, 65, 96)	0	block_1_ex
block_1_depthwise (DepthwiseConv2D)	(None, 32, 32, 96)	864	block_1_pa
block_1_depthwise_BN (BatchNormalization)	(None, 32, 32, 96)	384	block_1_de
block_1_depthwise_relu (ReLU)	(None, 32, 32, 96)	0	block_1_de
block_1_project (Conv2D)	(None, 32, 32, 24)	2,304	block_1_de
block_1_project_BN (BatchNormalization)	(None, 32, 32, 24)	96	block_1_pr
block_2_expand (Conv2D)	(None, 32, 32, 144)	3,456	block_1_pr
block_2_expand_BN (BatchNormalization)	(None, 32, 32, 144)	576	block_2_ex
block_2_expand_relu (ReLU)	(None, 32, 32, 144)	0	block_2_ex
block_2_depthwise (DepthwiseConv2D)	(None, 32, 32, 144)	1,296	block_2_ex
block_2_depthwise_BN (BatchNormalization)	(None, 32, 32, 144)	576	block_2_de
block_2_depthwise_relu (ReLU)	(None, 32, 32, 144)	0	block_2_de
block_2_project (Conv2D)	(None, 32, 32, 24)	3,456	block_2_de
block_2_project_BN (BatchNormalization)	(None, 32, 32, 24)	96	block_2_pr
block_2_add (Add)	(None, 32, 32, 24)	0	block_1_pr block_2_pr
block_3_expand (Conv2D)	(None, 32, 32, 144)	3,456	block_2_ad
block_3_expand_BN (BatchNormalization)	(None, 32, 32, 144)	576	block_3_ex
block_3_expand_relu (ReLU)	(None, 32, 32, 144)	0	block_3_ex
block_3_pad (ZeroPadding2D)	(None, 33, 33, 144)	0	block_3_ex
block_3_depthwise (DepthwiseConv2D)	(None, 16, 16, 144)	1,296	block_3_pa
block_3_depthwise_BN (BatchNormalization)	(None, 16, 16, 144)	576	block_3_de

block_3_depthwise_relu (ReLU)	(None, 16, 16, 144)	0	block_3_de
block_3_project (Conv2D)	(None, 16, 16, 32)	4,608	block_3_de
block_3_project_BN (BatchNormalization)	(None, 16, 16, 32)	128	block_3_pr
block_4_expand (Conv2D)	(None, 16, 16, 192)	6,144	block_3_pr
block_4_expand_BN (BatchNormalization)	(None, 16, 16, 192)	768	block_4_ex
block_4_expand_relu (ReLU)	(None, 16, 16, 192)	0	block_4_ex
block_4_depthwise (DepthwiseConv2D)	(None, 16, 16, 192)	1,728	block_4_ex
block_4_depthwise_BN (BatchNormalization)	(None, 16, 16, 192)	768	block_4_de
block_4_depthwise_relu (ReLU)	(None, 16, 16, 192)	0	block_4_de
block_4_project (Conv2D)	(None, 16, 16, 32)	6,144	block_4_de
block_4_project_BN (BatchNormalization)	(None, 16, 16, 32)	128	block_4_pr
block_4_add (Add)	(None, 16, 16, 32)	0	block_3_pr block_4_pr
block_5_expand (Conv2D)	(None, 16, 16, 192)	6,144	block_4_ad
block_5_expand_BN (BatchNormalization)	(None, 16, 16, 192)	768	block_5_ex
block_5_expand_relu (ReLU)	(None, 16, 16, 192)	0	block_5_ex
block_5_depthwise (DepthwiseConv2D)	(None, 16, 16, 192)	1,728	block_5_ex
block_5_depthwise_BN (BatchNormalization)	(None, 16, 16, 192)	768	block_5_de
block_5_depthwise_relu (ReLU)	(None, 16, 16, 192)	0	block_5_de
block_5_project (Conv2D)	(None, 16, 16, 32)	6,144	block_5_de
block_5_project_BN (BatchNormalization)	(None, 16, 16, 32)	128	block_5_pr
block_5_add (Add)	(None, 16, 16, 32)	0	block_4_ad block_5_pr
block_6_expand (Conv2D)	(None, 16, 16, 192)	6,144	block_5_ad
block_6_expand_BN (BatchNormalization)	(None, 16, 16, 192)	768	block_6_ex
block_6_expand_relu (ReLU)	(None, 16, 16, 192)	0	block_6_ex
block_6_pad (ZeroPadding2D)	(None, 17, 17, 192)	0	block_6_ex

block_6_depthwise (DepthwiseConv2D)	(None, 8, 8, 192)	1,728	block_6_pa
block_6_depthwise_BN (BatchNormalization)	(None, 8, 8, 192)	768	block_6_de
block_6_depthwise_relu (ReLU)	(None, 8, 8, 192)	0	block_6_de
block_6_project (Conv2D)	(None, 8, 8, 64)	12,288	block_6_de
block_6_project_BN (BatchNormalization)	(None, 8, 8, 64)	256	block_6_pr
block_7_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_6_pr
block_7_expand_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_7_ex
block_7_expand_relu (ReLU)	(None, 8, 8, 384)	0	block_7_ex
block_7_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_7_ex
block_7_depthwise_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_7_de
block_7_depthwise_relu (ReLU)	(None, 8, 8, 384)	0	block_7_de
block_7_project (Conv2D)	(None, 8, 8, 64)	24,576	block_7_de
block_7_project_BN (BatchNormalization)	(None, 8, 8, 64)	256	block_7_pr
block_7_add (Add)	(None, 8, 8, 64)	0	block_6_pr block_7_pr
block_8_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_7_ad
block_8_expand_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_8_ex
block_8_expand_relu (ReLU)	(None, 8, 8, 384)	0	block_8_ex
block_8_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_8_ex
block_8_depthwise_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_8_de
block_8_depthwise_relu (ReLU)	(None, 8, 8, 384)	0	block_8_de
block_8_project (Conv2D)	(None, 8, 8, 64)	24,576	block_8_de
block_8_project_BN (BatchNormalization)	(None, 8, 8, 64)	256	block_8_pr
block_8_add (Add)	(None, 8, 8, 64)	0	block_7_ad block_8_pr
block_9_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_8_ad

block_9_expand_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_9_ex
block_9_expand_relu (ReLU)	(None, 8, 8, 384)	0	block_9_ex
block_9_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_9_ex
block_9_depthwise_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_9_de
block_9_depthwise_relu (ReLU)	(None, 8, 8, 384)	0	block_9_de
block_9_project (Conv2D)	(None, 8, 8, 64)	24,576	block_9_de
block_9_project_BN (BatchNormalization)	(None, 8, 8, 64)	256	block_9_pr
block_9_add (Add)	(None, 8, 8, 64)	0	block_8_ad
block_9_add (Add)	(None, 8, 8, 64)	0	block_9_pr
block_10_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_9_ad
block_10_expand_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_10_e
block_10_expand_relu (ReLU)	(None, 8, 8, 384)	0	block_10_e
block_10_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_10_e
block_10_depthwise_BN (BatchNormalization)	(None, 8, 8, 384)	1,536	block_10_d
block_10_depthwise_relu (ReLU)	(None, 8, 8, 384)	0	block_10_d
block_10_project (Conv2D)	(None, 8, 8, 96)	36,864	block_10_d
block_10_project_BN (BatchNormalization)	(None, 8, 8, 96)	384	block_10_p
block_11_expand (Conv2D)	(None, 8, 8, 576)	55,296	block_10_p
block_11_expand_BN (BatchNormalization)	(None, 8, 8, 576)	2,304	block_11_e
block_11_expand_relu (ReLU)	(None, 8, 8, 576)	0	block_11_e
block_11_depthwise (DepthwiseConv2D)	(None, 8, 8, 576)	5,184	block_11_e
block_11_depthwise_BN (BatchNormalization)	(None, 8, 8, 576)	2,304	block_11_d
block_11_depthwise_relu (ReLU)	(None, 8, 8, 576)	0	block_11_d
block_11_project (Conv2D)	(None, 8, 8, 96)	55,296	block_11_d
block_11_project_BN (BatchNormalization)	(None, 8, 8, 96)	384	block_11_p

block_11_add (Add)	(None, 8, 8, 96)	0	block_10_p block_11_p
block_12_expand (Conv2D)	(None, 8, 8, 576)	55,296	block_11_a
block_12_expand_BN (BatchNormalization)	(None, 8, 8, 576)	2,304	block_12_e
block_12_expand_relu (ReLU)	(None, 8, 8, 576)	0	block_12_e
block_12_depthwise (DepthwiseConv2D)	(None, 8, 8, 576)	5,184	block_12_e
block_12_depthwise_BN (BatchNormalization)	(None, 8, 8, 576)	2,304	block_12_d
block_12_depthwise_relu (ReLU)	(None, 8, 8, 576)	0	block_12_d
block_12_project (Conv2D)	(None, 8, 8, 96)	55,296	block_12_d
block_12_project_BN (BatchNormalization)	(None, 8, 8, 96)	384	block_12_p
block_12_add (Add)	(None, 8, 8, 96)	0	block_11_a block_12_p
block_13_expand (Conv2D)	(None, 8, 8, 576)	55,296	block_12_a
block_13_expand_BN (BatchNormalization)	(None, 8, 8, 576)	2,304	block_13_e
block_13_expand_relu (ReLU)	(None, 8, 8, 576)	0	block_13_e
block_13_pad (ZeroPadding2D)	(None, 9, 9, 576)	0	block_13_e
block_13_depthwise (DepthwiseConv2D)	(None, 4, 4, 576)	5,184	block_13_p
block_13_depthwise_BN (BatchNormalization)	(None, 4, 4, 576)	2,304	block_13_d
block_13_depthwise_relu (ReLU)	(None, 4, 4, 576)	0	block_13_d
block_13_project (Conv2D)	(None, 4, 4, 160)	92,160	block_13_d
block_13_project_BN (BatchNormalization)	(None, 4, 4, 160)	640	block_13_p
block_14_expand (Conv2D)	(None, 4, 4, 960)	153,600	block_13_p
block_14_expand_BN (BatchNormalization)	(None, 4, 4, 960)	3,840	block_14_e
block_14_expand_relu (ReLU)	(None, 4, 4, 960)	0	block_14_e
block_14_depthwise (DepthwiseConv2D)	(None, 4, 4, 960)	8,640	block_14_e
block_14_depthwise_BN	(None, 4, 4, 960)	3,840	block_14_d

(BatchNormalization)			
block_14_depthwise_relu (ReLU)	(None, 4, 4, 960)	0	block_14_d
block_14_project (Conv2D)	(None, 4, 4, 160)	153,600	block_14_d
block_14_project_BN (BatchNormalization)	(None, 4, 4, 160)	640	block_14_p
block_14_add (Add)	(None, 4, 4, 160)	0	block_13_p block_14_p
block_15_expand (Conv2D)	(None, 4, 4, 960)	153,600	block_14_a
block_15_expand_BN (BatchNormalization)	(None, 4, 4, 960)	3,840	block_15_e
block_15_expand_relu (ReLU)	(None, 4, 4, 960)	0	block_15_e
block_15_depthwise (DepthwiseConv2D)	(None, 4, 4, 960)	8,640	block_15_e
block_15_depthwise_BN (BatchNormalization)	(None, 4, 4, 960)	3,840	block_15_d
block_15_depthwise_relu (ReLU)	(None, 4, 4, 960)	0	block_15_d
block_15_project (Conv2D)	(None, 4, 4, 160)	153,600	block_15_d
block_15_project_BN (BatchNormalization)	(None, 4, 4, 160)	640	block_15_p
block_15_add (Add)	(None, 4, 4, 160)	0	block_14_a block_15_p
block_16_expand (Conv2D)	(None, 4, 4, 960)	153,600	block_15_a
block_16_expand_BN (BatchNormalization)	(None, 4, 4, 960)	3,840	block_16_e
block_16_expand_relu (ReLU)	(None, 4, 4, 960)	0	block_16_e
block_16_depthwise (DepthwiseConv2D)	(None, 4, 4, 960)	8,640	block_16_e
block_16_depthwise_BN (BatchNormalization)	(None, 4, 4, 960)	3,840	block_16_d
block_16_depthwise_relu (ReLU)	(None, 4, 4, 960)	0	block_16_d
block_16_project (Conv2D)	(None, 4, 4, 320)	307,200	block_16_d
block_16_project_BN (BatchNormalization)	(None, 4, 4, 320)	1,280	block_16_p
Conv_1 (Conv2D)	(None, 4, 4, 1280)	409,600	block_16_p
Conv_1_bn (BatchNormalization)	(None, 4, 4, 1280)	5,120	Conv_1[0][1]

out_relu (ReLU)	(None, 4, 4, 1280)	0	Conv_1_bn[0]
global_average_pooling2d_5 (GlobalAveragePooling2D)	(None, 1280)	0	out_relu[0]

Total params: 2,257,984 (8.61 MB)

Trainable params: 2,223,872 (8.48 MB)

Non-trainable params: 34,112 (133.25 KB)

We can see that the MobileNet architecture has less number of trainable parameters compared to other architectures like VGGNet, ResNet, InceptionNet (GoogleNet), AlexNet.

```
In [78]: mobilenet_inc_model = Sequential(
    name='mobilenet_included_model',
    layers=[
        mobilenet_model,
        tf.keras.layers.Dense(units=1280, activation='relu', use_bias=True),
        tf.keras.layers.Dense(units=4, activation='softmax', use_bias=True)
    ]
)
mobilenet_inc_model.summary()
```

Model: "mobilenet\_included\_model"

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_128 (Functional)	?	2,257,984
dense_13 (Dense)	?	0 (unbuilt)
dense_14 (Dense)	?	0 (unbuilt)

Total params: 2,257,984 (8.61 MB)

Trainable params: 2,223,872 (8.48 MB)

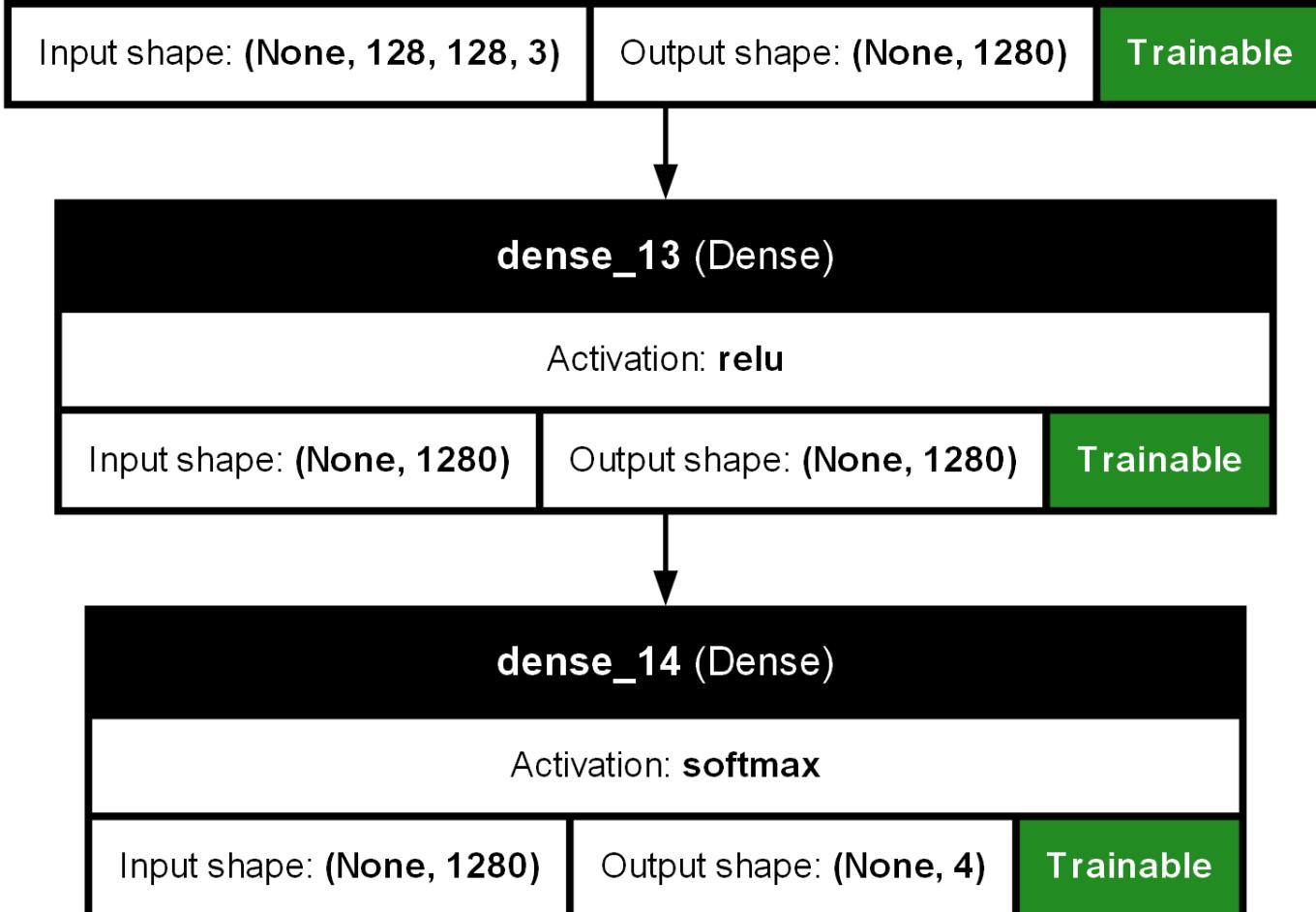
Non-trainable params: 34,112 (133.25 KB)

Plotting the model

```
In [149...]: tf.keras.utils.plot_model(mobilenet_inc_model, show_shapes=True, show_layer_names=True, sho
```

Out[149]:

## mobilenetv2\_1.00\_128 (Functional)



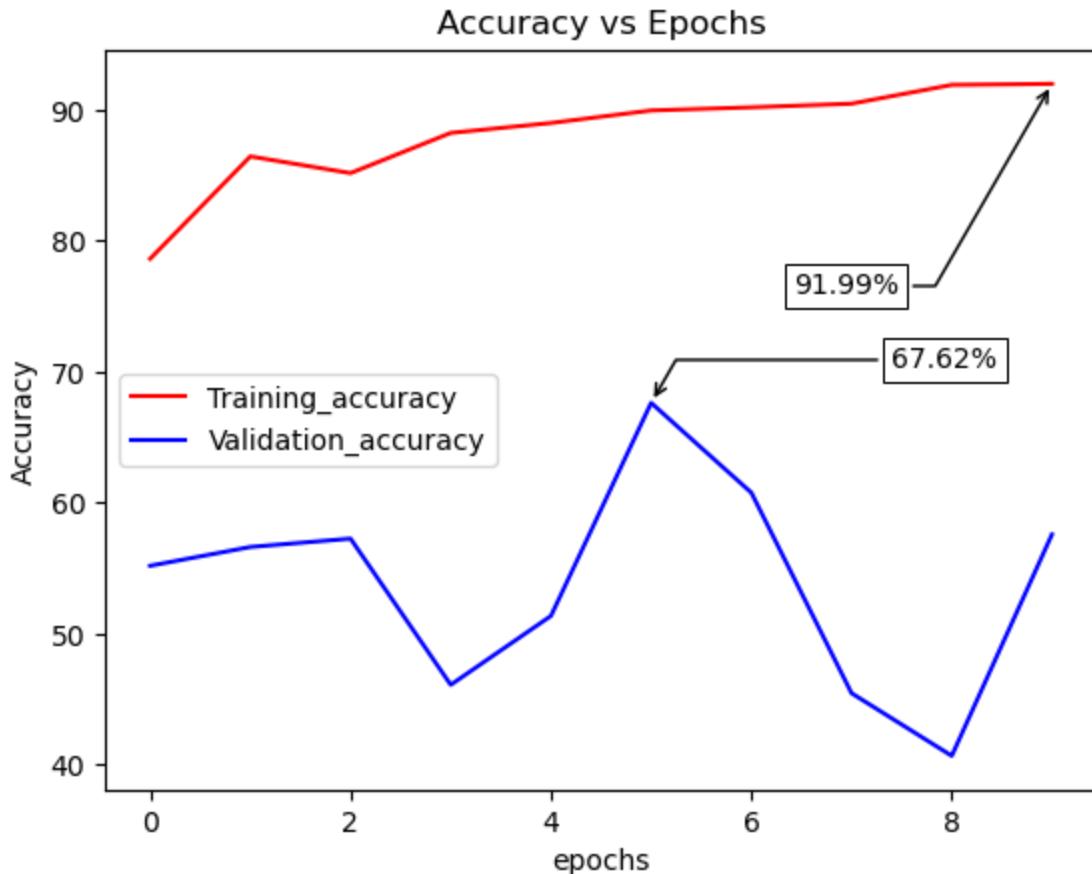
Compilation of Mobilenet model

```
In [79]: mobilenet_inc_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['a
mobilenet_direc = './mobilenet_model'
mobile_modelchk = tf.keras.callbacks.ModelCheckpoint(filepath=f'{mobilenet_direc}/model_
mobile_earlystop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', mode='min', min_de
mobile_tensorboard = tf.keras.callbacks.TensorBoard(log_dir=f'{mobilenet_direc}/model_
mobile_callbacks = [mobile_modelchk, mobile_earlystop, mobile_tensorboard]
fitted_mobilenet = mobilenet_inc_model.fit(aug_train_ds, validation_data=validation_ds, ep
Epoch 1/10
79/79 203s 2s/step - accuracy: 0.7193 - loss: 0.8404
- val_accuracy: 0.5518 - val_loss: 3.4870
Epoch 2/10
79/79 128s 2s/step - accuracy: 0.8673 - loss: 0.4362
- val_accuracy: 0.5662 - val_loss: 4.5182
Epoch 3/10
79/79 125s 2s/step - accuracy: 0.8651 - loss: 0.4498
- val_accuracy: 0.5726 - val_loss: 8.9030
Epoch 4/10
79/79 123s 2s/step - accuracy: 0.8835 - loss: 0.3469
- val_accuracy: 0.4609 - val_loss: 14.4858
Epoch 5/10
79/79 124s 2s/step - accuracy: 0.8895 - loss: 0.3120
- val_accuracy: 0.5136 - val_loss: 9.3757
Epoch 6/10
79/79 123s 2s/step - accuracy: 0.9081 - loss: 0.2475
- val_accuracy: 0.6762 - val_loss: 3.2831
Epoch 7/10
79/79 125s 2s/step - accuracy: 0.9042 - loss: 0.2727
```

```
- val_accuracy: 0.6077 - val_loss: 8.2643
Epoch 8/10
79/79 122s 2s/step - accuracy: 0.9154 - loss: 0.2437
- val_accuracy: 0.4545 - val_loss: 14.6695
Epoch 9/10
79/79 125s 2s/step - accuracy: 0.9171 - loss: 0.2401
- val_accuracy: 0.4067 - val_loss: 19.6958
Epoch 10/10
79/79 122s 2s/step - accuracy: 0.9243 - loss: 0.2226
- val_accuracy: 0.5758 - val_loss: 5.8057
```

### Training and Validation Accuracy

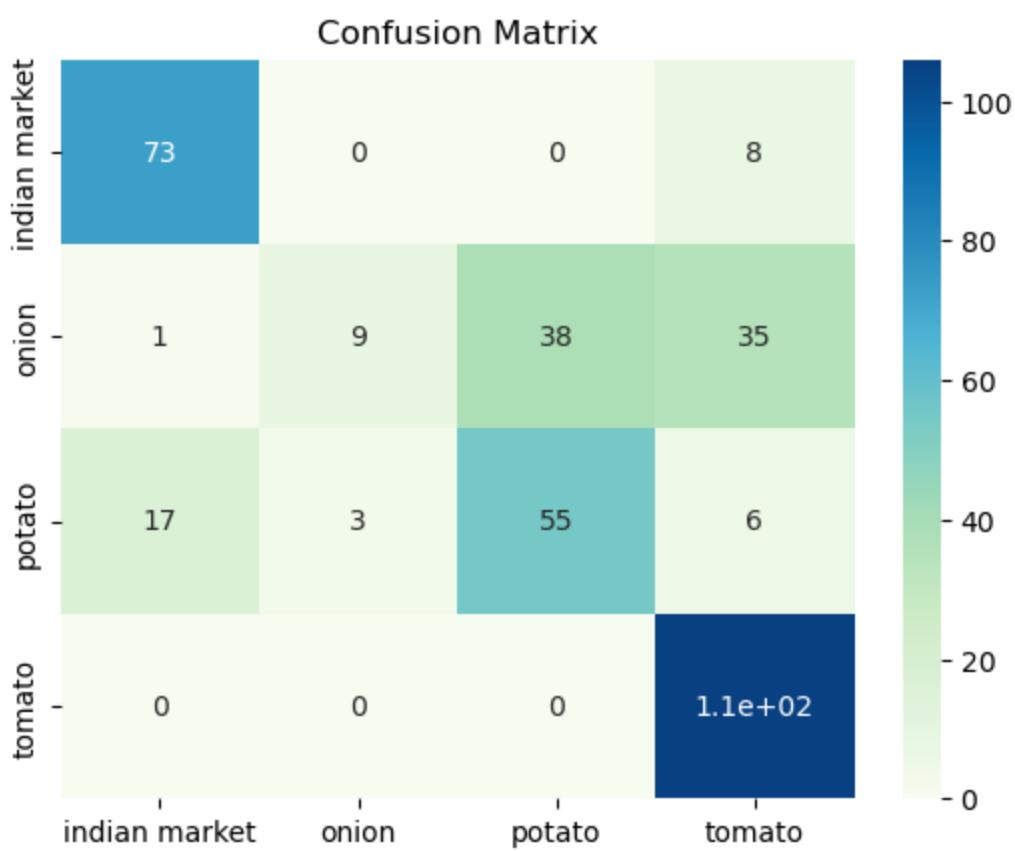
```
In [80]: plot_train_val_accuracy(fitted_mobilenet)
```



### Testing accuracy and Confusion matrix

```
In [81]: mobilenet_inc_model.load_weights(filepath=f'{mobilenet_dir}/model_chkpt/best_model.weights')
print_accuracy(mobilenet_inc_model, test_ds)
print()
print_confusion_matrix(mobilenet_inc_model, test_ds, show_plot=True)
```

The accuracy of the model on the provided dataset is 69.23%



```
In [82]: built_models['MobileNet'] = mobilenet_inc_model
```

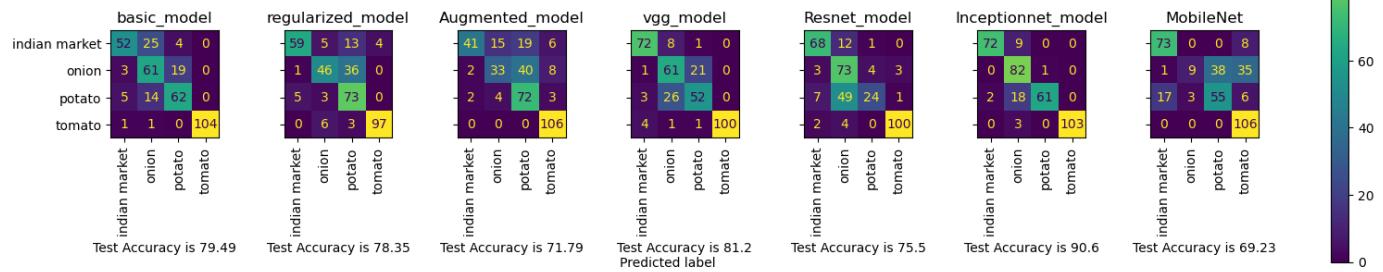
## Comparing the Models

```
In [145...]: from sklearn.metrics import ConfusionMatrixDisplay
f, axes = plt.subplots(1, len(built_models), figsize=(20,5), sharey='row')
for ind,model in enumerate(built_models):
    test_accuracy = print_accuracy(built_models[model],test_ds,show=False).round(2)
    conf_mat = print_confusion_matrix(built_models[model],test_ds,show_plot=False)
    disp = ConfusionMatrixDisplay(conf_mat,display_labels=test_data.class_names)
    disp.plot(ax=axes[ind], xticks_rotation=90)
    disp.ax_.set_title(f'{model}')
    disp.im_.colorbar.remove()
    disp.ax_.set_xlabel(f'Test Accuracy is {test_accuracy}')
    if i!=0:
        disp.ax_.set_ylabel('')

f.text(0.4, 0.1, 'Predicted label', ha='left')
plt.subplots_adjust(wspace=0.6, hspace=0.1)

f.colorbar(disp.im_, ax=axes)
plt.show()
```

```
Exception ignored in: <function WeakMethod.__new__.<locals>._cb at 0x0000018F450B0430>
Traceback (most recent call last):
  File "C:\Anaconda\lib\weakref.py", line 61, in __cb
    callback(self)
  File "C:\Anaconda\lib\site-packages\matplotlib\cbook\__init__.py", line 254, in _remove_proxy
    del self.callbacks[signal][cid]
KeyError: 'changed'
```



Out of all the models, Inception model turns out to give a good testing accuracy of 90.6%

### Model's performance on few random images

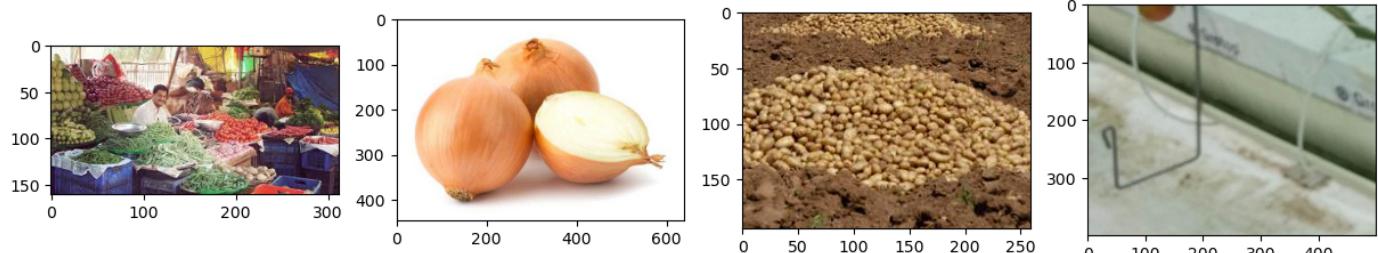
In [141...]

```
import random
images = []
for classes in test_data.class_names:
    directory = f'{main_directory}/test/{classes}'
    files = list(os.listdir(directory))
    img=random.choice(files)
    images.append(tf.keras.preprocessing.image.load_img(f'{directory}/{img}'))
```

plt.figure(figsize=(15,5))
for i in range(len(images)):
 plt.suptitle(f'5 Samples from different classes')
 plt.subplot(1,4,i+1)
 plt.imshow(images[i])

plt.show()

5 Samples from different classes



Sending these Random images into the Model and predicting its class

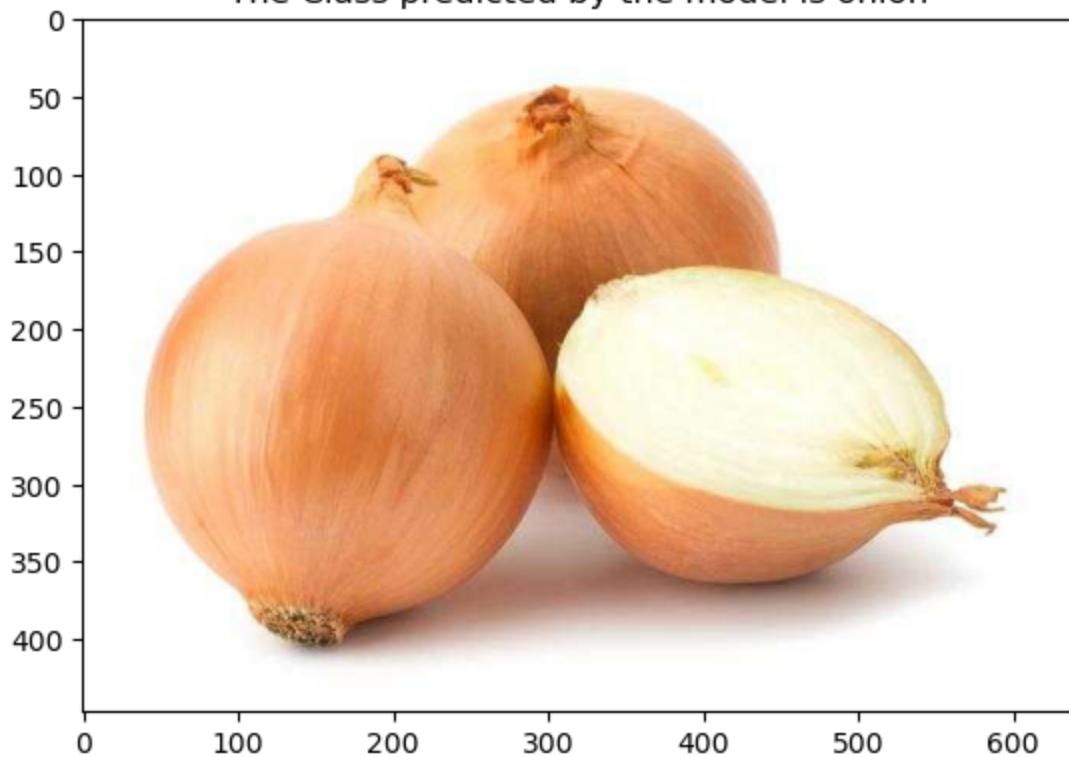
In [142...]

```
for img in images:
    orig_img = img
    img = tf.keras.preprocessing.image.smart_resize(img, (128,128))
    img = tf.keras.preprocessing.image.img_to_array(img)
    img = img/255
    img = np.expand_dims(img, axis=0)
    best_model = built_models['Inceptionnet_model']
    prediction = best_model.predict(img, verbose=0)
    predicted_label = np.argmax(prediction)
    prediction_labels = {0:'indian market',1:'onion',2:'potato',3:'tomato'}
    plt.title(f'The Class predicted by the model is {prediction_labels[predicted_label]}')
    plt.imshow(orig_img)
    plt.show()
    print('='*100)
```

The Class predicted by the model is onion



The Class predicted by the model is onion



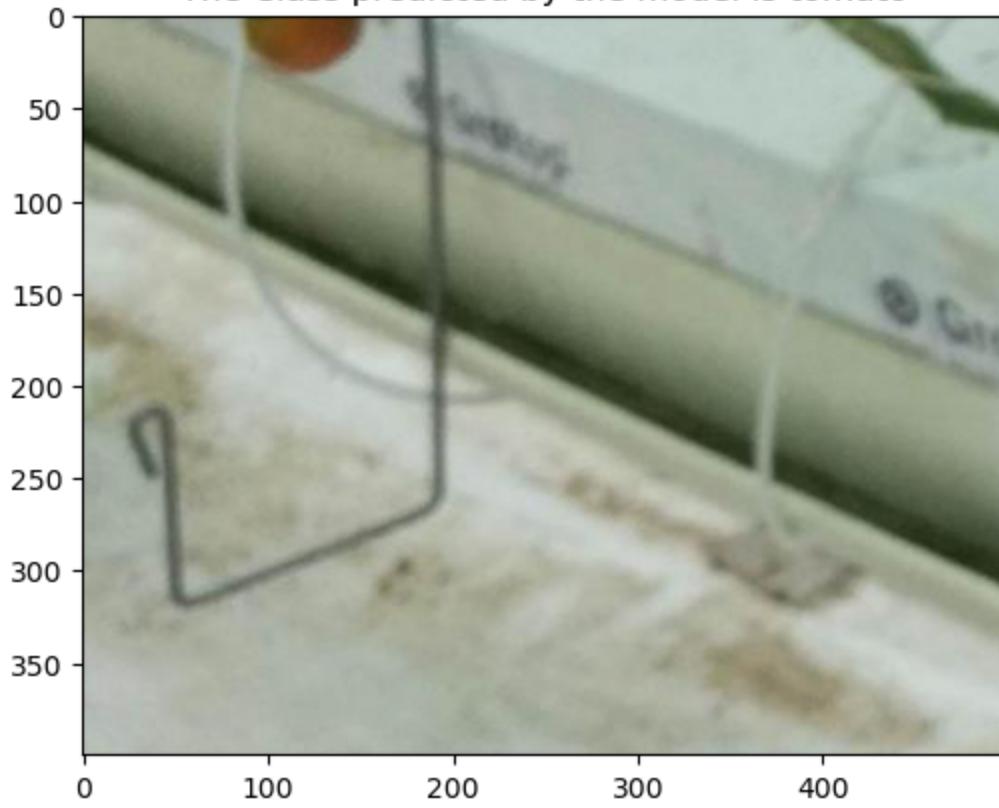
The Class predicted by the model is potato



=====

=====

The Class predicted by the model is tomato



=====

=====

Summary:

The model's accuracy can further be improved by Training certain layers of the transfer learning model or by including more dense layers at the top of the built network.

## Saving the best model for easy export and import

```
In [144]: built_models['Inceptionnet_model'].save("./best_model_ninjacart/inception_model.keras")
```