101 Food Classification & Calories Identification

Classes & Calories

-The calories per gram for each food item:

- **Apple Pie: ~2.5 calories per gram**
- **Baby Back Ribs: ~3.5 calories per gram**
- **Baklava: ~5 calories per gram**
- **Beef Carpaccio: ~2 calories per gram**
- **Beef Tartare: ~2.5 calories per gram**
- **Beet Salad: ~0.5 calories per gram**
- **Beignets: ~3.5 calories per gram**
- **Bibimbap: ~1.5 calories per gram**
- **Bread Pudding: ~2.5 calories per gram**
- **Breakfast Burrito: ~2 calories per gram**
- **Bruschetta: ~1 calorie per gram**
- **Caesar Salad: ~0.5 calories per gram**
- **Cannoli: ~3.5 calories per gram**
- **Caprese Salad: ~1 calorie per gram**
- **Carrot Cake: ~3.5 calories per gram**
- **Ceviche: ~0.5 calories per gram**
- **Cheese Plate: ~3.5 calories per gram**
- **Cheesecake: ~3.5 calories per gram**
- **Chicken Curry: ~1.5 calories per gram**
- **Chicken Quesadilla: ~2.5 calories per gram**
- **Chicken Wings: ~3 calories per gram**
- **Chocolate Cake: ~4 calories per gram**
- **Chocolate Mousse: ~3 calories per gram**
- **Churros: ~4 calories per gram**
- **Clam Chowder: ~1.5 calories per gram**
- **Club Sandwich: ~2.5 calories per gram**
- **Crab Cakes: ~2 calories per gram**
- **Creme Brulee: ~3.5 calories per gram**
- **Croque Madame: ~3 calories per gram**
- **Cupcakes: ~3.5 calories per gram**
- **Deviled Eggs: ~1 calorie per gram**
- **Donuts: ~4 calories per gram**
- **Dumplings: ~2.5 calories per gram**
- **Edamame: ~1 calorie per gram**
- **Eggs Benedict: ~2.5 calories per gram**
- **Escargots: ~1 calorie per gram**
- **Falafel: ~2 calories per gram**
- **Filet Mignon: ~2.5 calories per gram**
- **Fish and Chips: ~2.5 calories per gram**
- **Foie Gras: ~4.5 calories per gram**
- **French Fries: ~3.5 calories per gram**
- **French Onion Soup: ~1 calorie per gram**
- **French Toast: ~2 calories per gram**
- **Fried Calamari: ~2.5 calories per gram**
- **Fried Rice: ~1.5 calories per gram**
- **Frozen Yogurt: ~1 calorie per gram**
- **Garlic Bread: ~4 calories per gram**

- **Gnocchi: ~1.5 calories per gram**
- **Greek Salad: ~0.5 calories per gram**
- **Grilled Cheese Sandwich: ~3 calories per gram**
- **Grilled Salmon: ~2 calories per gram**
- **Guacamole: ~2 calories per gram**
- **Gyoza: ~2 calories per gram**
- **Hamburger: ~3.5 calories per gram**
- **Hot and Sour Soup: ~0.5 calories per gram**
- **Hot Dog: ~3.5 calories per gram**
- **Huevos Rancheros: ~2 calories per gram**
- **Hummus: ~1.5 calories per gram**
- **Ice Cream: ~2 calories per gram**
- **Lasagna: ~1.5 calories per gram**
- **Lobster Bisque: ~1 calorie per gram**
- **Lobster Roll Sandwich: ~2.5 calories per gram**
- **Macaroni and Cheese: ~3 calories per gram**
- **Macarons: ~4 calories per gram**
- **Miso Soup: ~0.5 calories per gram**
- **Mussels: ~0.5 calories per gram**
- **Nachos: ~2.5 calories per gram**
- **Omelette: ~1.5 calories per gram**
- **Onion Rings: ~2.5 calories per gram**
- **Oysters: ~0.5 calories per gram**
- **Pad Thai: ~2 calories per gram**
- **Paella: ~1.5 calories per gram**
- **Pancakes: ~2 calories per gram**
- **Panna Cotta: ~3.5 calories per gram**
- **Peking Duck: ~4 calories per gram**
- **Pho: ~1 calorie per gram**
- **Pizza: ~2.5 calories per gram**
- **Pork Chop: ~2.5 calories per gram**
- **Poutine: ~2.5 calories per gram**
- **Prime Rib: ~2.5 calories per gram**
- **Pulled Pork Sandwich: ~2.5 calories per gram**
- **Ramen: ~1 calorie per gram**
- **Ravioli: ~1.5 calories per gram**
- **Red Velvet Cake: ~4 calories per gram**
- **Risotto: ~1.5 calories per gram**
- **Samosa: ~2 calories per gram**
- **Sashimi: ~1 calorie per gram**
- **Scallops: ~1 calorie per gram**
- **Seaweed Salad: ~0.5 calories per gram**
- **Shrimp and Grits: ~2 calories per gram**
- **Spaghetti Bolognese: ~1.5 calories per gram**
- **Spaghetti Carbonara: ~2 calories per gram**
- **Spring Rolls: ~1.5 calories per gram**
- **Steak: ~2.5 calories per gram**
- **Strawberry Shortcake: ~3.5 calories per gram**
- **Sushi: ~1 calorie per gram**
- **Tacos: ~2 calories per gram**
- **Takoyaki: ~2.5 calories per gram**
- **Tiramisu: ~3 calories per gram**
- **Tuna Tartare: ~1.5 calories per gram**
- **Waffles: ~2 calories per gram**

In [ ]:

```
!pip install keras==2.15.0
```

```
Requirement already satisfied: keras==2.15.0 in /opt/conda/lib/python3.10/site-packages (
2.15.0)
```

In [ ]:

```
import keras
keras.__version__
```

Out[ ]:

```
'2.15.0'
```

In [ ]:

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sn
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import vgg16
from keras.src.layers.pooling.average_pooling2d import AvgPool2D
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
from keras.models import Sequential
from keras.layers import Dense,Input,Flatten
from tensorflow.keras.utils import load_img,img_to_array
from sklearn.metrics import confusion_matrix
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
```

In [ ]:

```
import os
print(os.listdir("/kaggle/input/food-101/food-101/food-101/images/"))
```

```
['macarons', 'french_toast', 'lobster_bisque', 'prime_rib', 'pork_chop', 'guacamole', 'ba
by_back_ribs', 'mussels', 'beef_carpaccio', 'poutine', 'hot_and_sour_soup', 'seaweed_sala
d', 'foie_gras', 'dumplings', 'peking_duck', 'takoyaki', 'bibimbap', 'falafel', 'pulled_p
ork_sandwich', 'lobster_roll_sandwich', 'carrot_cake', 'beet_salad', 'panna_cotta', 'donu
ts', 'red_velvet_cake', 'grilled_cheese_sandwich', 'cannoli', 'spring_rolls', 'shrimp_and
_grits', 'clam_chowder', 'omelette', 'fried_calamari', 'caprese_salad', 'oysters', 'scall
ops', 'ramen', 'grilled_salmon', 'croque_madame', 'filet_mignon', 'hamburger', 'spaghetti
_carbonara', 'miso_soup', 'bread_pudding', 'lasagna', 'crab_cakes', 'cheesecake', 'spaghe
tti_bolognese', 'cup_cakes', 'creme_brulee', 'waffles', 'fish_and_chips', 'paella', 'maca
roni_and_cheese', 'chocolate_mousse', 'ravioli', 'chicken_curry', 'caesar_salad', 'nachos
', 'tiramisu', 'frozen_yogurt', 'ice_cream', 'risotto', 'club_sandwich', 'strawberry_shor
tcake', 'steak', 'churros', 'garlic_bread', 'baklava', 'bruschetta', 'hummus', 'chicken_w
ings', 'greek_salad', 'tuna_tartare', 'chocolate_cake', 'gyoza', 'eggs_benedict', 'devile
d_eggs', 'samosa', 'sushi', 'breakfast_burrito', 'ceviche', 'beef_tartare', 'apple_pie',
'.DS_Store', 'huevos_rancheros', 'beignets', 'pizza', 'edamame', 'french_onion_soup', 'ho
t_dog', 'tacos', 'chicken_quesadilla', 'pho', 'gnocchi', 'pancakes', 'fried_rice', 'chees
e_plate', 'onion_rings', 'escargots', 'sashimi', 'pad_thai', 'french_fries']
```

Food Classes

In [ ]:

```
values = ['macarons', 'french_toast', 'lobster_bisque', 'prime_rib', 'pork_chop', 'guaca
mole', 'baby_back_ribs', 'mussels', 'beef_carpaccio', 'poutine', 'hot_and_sour_soup', 'se
aweed_salad', 'foie_gras', 'dumplings', 'peking_duck', 'takoyaki', 'bibimbap', 'falafel'
, 'pulled_pork_sandwich', 'lobster_roll_sandwich', 'carrot_cake', 'beet_salad', 'panna_co
tta', 'donuts', 'red_velvet_cake', 'grilled_cheese_sandwich', 'cannoli', 'spring_rolls',
'shrimp_and_grits', 'clam_chowder', 'omelette', 'fried_calamari', 'caprese_salad', 'oyste
rs', 'scallops', 'ramen', 'grilled_salmon', 'croque_madame', 'filet_mignon', 'hamburger'
, 'spaghetti_carbonara', 'miso_soup', 'bread_pudding', 'lasagna', 'crab_cakes', 'cheesec
ake', 'spaghetti_bolognese', 'cup_cakes', 'creme_brulee', 'waffles', 'fish_and_chips', '
paella', 'macaroni_and_cheese', 'chocolate_mousse', 'ravioli', 'chicken_curry', 'caesar_s
alad', 'nachos', 'tiramisu', 'frozen_yogurt', 'ice_cream', 'risotto', 'club_sandwich', '
```

```
strawberry_shortcake', 'steak', 'churros', 'garlic_bread', 'baklava', 'bruschetta', 'hummus', 'chicken_wings', 'greek_salad', 'tuna_tartare', 'chocolate_cake', 'gyoza', 'eggs_benedict', 'deviled_eggs', 'samosa', 'sushi', 'breakfast_burrito', 'ceviche', 'beef_tartare', 'apple_pie', '.DS_Store', 'huevos_rancheros', 'beignets', 'pizza', 'edamame', 'french_onion_soup', 'hot_dog', 'tacos', 'chicken_quesadilla', 'pho', 'gnocchi', 'pancakes', 'fried_rice', 'cheese_plate', 'onion_rings', 'escargots', 'sashimi', 'pad_thai', 'french_fries']
values.sort()
values = values[1:]
print(values)
```

```
['apple_pie', 'baby_back_ribs', 'baklava', 'beef_carpaccio', 'beef_tartare', 'beet_salad', 'beignets', 'bibimbap', 'bread_pudding', 'breakfast_burrito', 'bruschetta', 'caesar_salad', 'cannoli', 'caprese_salad', 'carrot_cake', 'ceviche', 'cheese_plate', 'cheesecake', 'chicken_curry', 'chicken_quesadilla', 'chicken_wings', 'chocolate_cake', 'chocolate_mousse', 'churros', 'clam_chowder', 'club_sandwich', 'crab_cakes', 'creme_brulee', 'croque_madame', 'cup_cakes', 'deviled_eggs', 'donuts', 'dumplings', 'edamame', 'eggs_benedict', 'escargots', 'falafel', 'filet_mignon', 'fish_and_chips', 'foie_gras', 'french_fries', 'french_onion_soup', 'french_toast', 'fried_calamari', 'fried_rice', 'frozen_yogurt', 'garlic_bread', 'gnocchi', 'greek_salad', 'grilled_cheese_sandwich', 'grilled_salmon', 'guacamole', 'gyoza', 'hamburger', 'hot_and_sour_soup', 'hot_dog', 'huevos_rancheros', 'hummus', 'ice_cream', 'lasagna', 'lobster_bisque', 'lobster_roll_sandwich', 'macaroni_and_cheese', 'macarons', 'miso_soup', 'mussels', 'nachos', 'omelette', 'onion_rings', 'oysters', 'pad_thai', 'paella', 'pancakes', 'panna_cotta', 'peking_duck', 'pho', 'pizza', 'pork_chop', 'poutine', 'prime_rib', 'pulled_pork_sandwich', 'ramen', 'ravioli', 'red_velvet_cake', 'risotto', 'samosa', 'sashimi', 'scallops', 'seaweed_salad', 'shrimp_and_grits', 'spaghetti_bolognese', 'spaghetti_carbonara', 'spring_rolls', 'steak', 'strawberry_shortcake', 'sushi', 'tacos', 'takoyaki', 'tiramisu', 'tuna_tartare', 'waffles']
```

In [ ]:

```
print("Number of classes:",len(values))
```

```
Number of classes: 101
```

In [ ]:

```
s = """Apple Pie: ~2.5 calories per gram
Baby Back Ribs: ~3.5 calories per gram
Baklava: ~5 calories per gram
Beef Carpaccio: ~2 calories per gram
Beef Tartare: ~2.5 calories per gram
Beet Salad: ~0.5 calories per gram
Beignets: ~3.5 calories per gram
Bibimbap: ~1.5 calories per gram
Bread Pudding: ~2.5 calories per gram
Breakfast Burrito: ~2 calories per gram
Bruschetta: ~1 calorie per gram
Caesar Salad: ~0.5 calories per gram
Cannoli: ~3.5 calories per gram
Caprese Salad: ~1 calorie per gram
Carrot Cake: ~3.5 calories per gram
Ceviche: ~0.5 calories per gram
Cheese Plate: ~3.5 calories per gram
Cheesecake: ~3.5 calories per gram
Chicken Curry: ~1.5 calories per gram
Chicken Quesadilla: ~2.5 calories per gram
Chicken Wings: ~3 calories per gram
Chocolate Cake: ~4 calories per gram
Chocolate Mousse: ~3 calories per gram
Churros: ~4 calories per gram
Clam Chowder: ~1.5 calories per gram
Club Sandwich: ~2.5 calories per gram
Crab Cakes: ~2 calories per gram
Creme Brulee: ~3.5 calories per gram
Croque Madame: ~3 calories per gram
Cupcakes: ~3.5 calories per gram
Deviled Eggs: ~1 calorie per gram
Donuts: ~4 calories per gram
Dumplings: ~2.5 calories per gram
Edamame: ~1 calorie per gram
Eggs Benedict: ~2.5 calories per gram
```

```
Escargots: ~1 calorie per gram
Falafel: ~2 calories per gram
Filet Mignon: ~2.5 calories per gram
Fish and Chips: ~2.5 calories per gram
Foie Gras: ~4.5 calories per gram
French Fries: ~3.5 calories per gram
French Onion Soup: ~1 calorie per gram
French Toast: ~2 calories per gram
Fried Calamari: ~2.5 calories per gram
Fried Rice: ~1.5 calories per gram
Frozen Yogurt: ~1 calorie per gram
Garlic Bread: ~4 calories per gram
Gnocchi: ~1.5 calories per gram
Greek Salad: ~0.5 calories per gram
Grilled Cheese Sandwich: ~3 calories per gram
Grilled Salmon: ~2 calories per gram
Guacamole: ~2 calories per gram
Gyoza: ~2 calories per gram
Hamburger: ~3.5 calories per gram
Hot and Sour Soup: ~0.5 calories per gram
Hot Dog: ~3.5 calories per gram
Huevos Rancheros: ~2 calories per gram
Hummus: ~1.5 calories per gram
Ice Cream: ~2 calories per gram
Lasagna: ~1.5 calories per gram
Lobster Bisque: ~1 calorie per gram
Lobster Roll Sandwich: ~2.5 calories per gram
Macaroni and Cheese: ~3 calories per gram
Macarons: ~4 calories per gram
Miso Soup: ~0.5 calories per gram
Mussels: ~0.5 calories per gram
Nachos: ~2.5 calories per gram
Omelette: ~1.5 calories per gram
Onion Rings: ~2.5 calories per gram
Oysters: ~0.5 calories per gram
Pad Thai: ~2 calories per gram
Paella: ~1.5 calories per gram
Pancakes: ~2 calories per gram
Panna Cotta: ~3.5 calories per gram
Peking Duck: ~4 calories per gram
Pho: ~1 calorie per gram
Pizza: ~2.5 calories per gram
Pork Chop: ~2.5 calories per gram
Poutine: ~2.5 calories per gram
Prime Rib: ~2.5 calories per gram
Pulled Pork Sandwich: ~2.5 calories per gram
Ramen: ~1 calorie per gram
Ravioli: ~1.5 calories per gram
Red Velvet Cake: ~4 calories per gram
Risotto: ~1.5 calories per gram
Samosa: ~2 calories per gram
Sashimi: ~1 calorie per gram
Scallops: ~1 calorie per gram
Seaweed Salad: ~0.5 calories per gram
Shrimp and Grits: ~2 calories per gram
Spaghetti Bolognese: ~1.5 calories per gram
Spaghetti Carbonara: ~2 calories per gram
Spring Rolls: ~1.5 calories per gram
Steak: ~2.5 calories per gram
Strawberry Shortcake: ~3.5 calories per gram
Sushi: ~1 calorie per gram
Tacos: ~2 calories per gram
Takoyaki: ~2.5 calories per gram
Tiramisu: ~3 calories per gram
Tuna Tartare: ~1.5 calories per gram
Waffles: ~2 calories per gram
"""
calories = s.splitlines()
s = "These values are approximations and can vary based on factors such as ingredients an
d cooking methods."
```

In [ ]:

In [ ]:

```
len(calories)
```

Out[ ]:

```
101
```

In [ ]:

```
calories[0]
```

Out[ ]:

```
'Apple Pie: ~2.5 calories per gram'
```

In [ ]:

```
print("First element:",values[0],"\nLast element:",values[-1])
```

```
First element: apple_pie
Last element: waffles
```

In [ ]:

```
train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.1)
train_data = train_datagen.flow_from_directory('/kaggle/input/food-101/food-101/food-101/
images/',
                                               target_size=(224,224),
                                               batch_size=100,
                                               class_mode='categorical',
                                               shuffle=True,
                                               subset='training')
test_data = train_datagen.flow_from_directory('/kaggle/input/food-101/food-101/food-101/i
mages/',
                                               target_size=(224,224),
                                               batch_size=100,
                                               class_mode='categorical',
                                               shuffle=False,
                                               subset='validation')
```

```
Found 90900 images belonging to 101 classes.
Found 10100 images belonging to 101 classes.
```

In [ ]:

```
print("Images Shape:",train_data.image_shape)
```

```
Images Shape: (224, 224, 3)
```

In [ ]:

```
print('\nBatch Size:',100,
     "\nNunmber of Batches in training set:",len(train_data),
     "\nNunmber of Batches in testing set:",len(test_data),
     "\nNumber of Samples in training set:",train_data.samples,"Samples",
     "\nNumber of Samples in testing set:",test_data.samples,"Samples")
```

```
Batch Size: 100
Nunmber of Batches in training set: 909
Nunmber of Batches in testing set: 101
Number of Samples in training set: 90900 Samples
Number of Samples in testing set: 10100 Samples
```

In [ ]:

```
print("\nThe 101 Classes numbers:\n",np.unique(train_data.labels),"\n",
      "_"*30,
      "\nThe 101 classes names:\n",train_data.class_indices,
     sep="")
```

```
The 101 Classes numbers:
[  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
```

```
 36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
 54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
 72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
 90  91  92  93  94  95  96  97  98  99 100]
______________________________
The 101 classes names:
{'apple_pie': 0, 'baby_back_ribs': 1, 'baklava': 2, 'beef_carpaccio': 3, 'beef_tartare':
4, 'beet_salad': 5, 'beignets': 6, 'bibimbap': 7, 'bread_pudding': 8, 'breakfast_burrito'
: 9, 'bruschetta': 10, 'caesar_salad': 11, 'cannoli': 12, 'caprese_salad': 13, 'carrot_ca
ke': 14, 'ceviche': 15, 'cheese_plate': 16, 'cheesecake': 17, 'chicken_curry': 18, 'chick
en_quesadilla': 19, 'chicken_wings': 20, 'chocolate_cake': 21, 'chocolate_mousse': 22, 'c
hurros': 23, 'clam_chowder': 24, 'club_sandwich': 25, 'crab_cakes': 26, 'creme_brulee': 2
7, 'croque_madame': 28, 'cup_cakes': 29, 'deviled_eggs': 30, 'donuts': 31, 'dumplings': 3
2, 'edamame': 33, 'eggs_benedict': 34, 'escargots': 35, 'falafel': 36, 'filet_mignon': 37
, 'fish_and_chips': 38, 'foie_gras': 39, 'french_fries': 40, 'french_onion_soup': 41, 'fr
ench_toast': 42, 'fried_calamari': 43, 'fried_rice': 44, 'frozen_yogurt': 45, 'garlic_bre
ad': 46, 'gnocchi': 47, 'greek_salad': 48, 'grilled_cheese_sandwich': 49, 'grilled_salmon
': 50, 'guacamole': 51, 'gyoza': 52, 'hamburger': 53, 'hot_and_sour_soup': 54, 'hot_dog':
55, 'huevos_rancheros': 56, 'hummus': 57, 'ice_cream': 58, 'lasagna': 59, 'lobster_bisque
': 60, 'lobster_roll_sandwich': 61, 'macaroni_and_cheese': 62, 'macarons': 63, 'miso_soup
': 64, 'mussels': 65, 'nachos': 66, 'omelette': 67, 'onion_rings': 68, 'oysters': 69, 'pa
d_thai': 70, 'paella': 71, 'pancakes': 72, 'panna_cotta': 73, 'peking_duck': 74, 'pho': 7
5, 'pizza': 76, 'pork_chop': 77, 'poutine': 78, 'prime_rib': 79, 'pulled_pork_sandwich':
80, 'ramen': 81, 'ravioli': 82, 'red_velvet_cake': 83, 'risotto': 84, 'samosa': 85, 'sash
imi': 86, 'scallops': 87, 'seaweed_salad': 88, 'shrimp_and_grits': 89, 'spaghetti_bologne
se': 90, 'spaghetti_carbonara': 91, 'spring_rolls': 92, 'steak': 93, 'strawberry_shortcak
e': 94, 'sushi': 95, 'tacos': 96, 'takoyaki': 97, 'tiramisu': 98, 'tuna_tartare': 99, 'wa
ffles': 100}
```

Model - Inception V3 Architecture

In [ ]:

```
from keras.applications.inception_v3 import InceptionV3
from keras.applications.inception_v3 import preprocess_input, decode_predictions
from keras.preprocessing import image
from keras.layers import Input
from keras.models import Sequential, Model
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D, MaxPooling2D, ZeroPadding2D, GlobalAveragePooling
2D, AveragePooling2D
from keras.callbacks import ModelCheckpoint, CSVLogger, LearningRateScheduler, ReduceLROn
Plateau
from keras.optimizers import SGD
from keras.regularizers import l2
import keras.backend as K
import math
```

In [ ]:

```
base_model3 = InceptionV3(weights='imagenet', include_top=False, input_tensor=Input(shap
e=(224, 224, 3)))

x = base_model3.output
x = AveragePooling2D()(x)
x = Dropout(.5)(x)
x = Flatten()(x)
x = Dense(101, kernel_initializer='glorot_uniform', kernel_regularizer=l2(.0005), activa
tion='softmax')(x)
model3 = Model(inputs=base_model3.input, outputs=x)
model3.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________
_________
 Layer (type)                   Output Shape         Param #     Connected to

==========================================================================================
=========
 input_15 (InputLayer)          [(None, 224, 224, 3)]  0          []
```

```
 input_15 (InputLayer)       [(None, 224, 224, 3)]        0           []


 conv2d_1034 (Conv2D)        (None, 111, 111, 32)         864         ['input_15[0][0]']


 batch_normalization_1034 (  (None, 111, 111, 32)         96          ['conv2d_1034[0][0]']

 BatchNormalization)


 activation_1034 (Activatio  (None, 111, 111, 32)         0           ['batch_normalizatio
n_1034[0][
 n)                                                                   0]']


 conv2d_1035 (Conv2D)        (None, 109, 109, 32)         9216        ['activation_1034[0]
[0]']

 batch_normalization_1035 (  (None, 109, 109, 32)         96          ['conv2d_1035[0][0]']

 BatchNormalization)


 activation_1035 (Activatio  (None, 109, 109, 32)         0           ['batch_normalizatio
n_1035[0][
 n)                                                                   0]']


 conv2d_1036 (Conv2D)        (None, 109, 109, 64)         18432       ['activation_1035[0]
[0]']

 batch_normalization_1036 (  (None, 109, 109, 64)         192         ['conv2d_1036[0][0]']

 BatchNormalization)


 activation_1036 (Activatio  (None, 109, 109, 64)         0           ['batch_normalizatio
n_1036[0][
 n)                                                                   0]']


 max_pooling2d_44 (MaxPooli  (None, 54, 54, 64)           0           ['activation_1036[0]
[0]']
 ng2D)


 conv2d_1037 (Conv2D)        (None, 54, 54, 80)           5120        ['max_pooling2d_44[0
][0]']

 batch_normalization_1037 (  (None, 54, 54, 80)           240         ['conv2d_1037[0][0]']

 BatchNormalization)


 activation_1037 (Activatio  (None, 54, 54, 80)           0           ['batch_normalizatio
n_1037[0][
 n)                                                                   0]']
```

```
 conv2d_1038 (Conv2D)          (None, 52, 52, 192)          138240     ['activation_1037[0]
[0]']

 batch_normalization_1038 (   (None, 52, 52, 192)          576        ['conv2d_1038[0][0]']

 BatchNormalization)


 activation_1038 (Activatio   (None, 52, 52, 192)          0          ['batch_normalizatio
n_1038[0][
 n)                                                                   0]']


 max_pooling2d_45 (MaxPooli   (None, 25, 25, 192)          0          ['activation_1038[0]
[0]']
 ng2D)


 conv2d_1042 (Conv2D)          (None, 25, 25, 64)           12288      ['max_pooling2d_45[0
][0]']

 batch_normalization_1042 (   (None, 25, 25, 64)           192        ['conv2d_1042[0][0]']

 BatchNormalization)


 activation_1042 (Activatio   (None, 25, 25, 64)           0          ['batch_normalizatio
n_1042[0][
 n)                                                                   0]']


 conv2d_1040 (Conv2D)          (None, 25, 25, 48)           9216       ['max_pooling2d_45[0
][0]']

 conv2d_1043 (Conv2D)          (None, 25, 25, 96)           55296      ['activation_1042[0]
[0]']

 batch_normalization_1040 (   (None, 25, 25, 48)           144        ['conv2d_1040[0][0]']

 BatchNormalization)


 batch_normalization_1043 (   (None, 25, 25, 96)           288        ['conv2d_1043[0][0]']

 BatchNormalization)


 activation_1040 (Activatio   (None, 25, 25, 48)           0          ['batch_normalizatio
n_1040[0][
 n)                                                                   0]']


 activation_1043 (Activatio   (None, 25, 25, 96)           0          ['batch_normalizatio
n_1043[0][
 n)                                                                   0]']


 average_pooling2d_109 (Ave   (None, 25, 25, 192)          0          ['max_pooling2d_45[0
```

```
 average_pooling2d_109 (Ave  (None, 25, 25, 192)          0         ['max_pooling2d_45[0
][0]']
 ragePooling2D)


 conv2d_1039 (Conv2D)        (None, 25, 25, 64)           12288     ['max_pooling2d_45[0
][0]']

 conv2d_1041 (Conv2D)        (None, 25, 25, 64)           76800     ['activation_1040[0]
[0]']

 conv2d_1044 (Conv2D)        (None, 25, 25, 96)           82944     ['activation_1043[0]
[0]']

 conv2d_1045 (Conv2D)        (None, 25, 25, 32)           6144      ['average_pooling2d_
109[0][0]'
                                                                    ]

 batch_normalization_1039 (  (None, 25, 25, 64)           192       ['conv2d_1039[0][0]']
 BatchNormalization)

 batch_normalization_1041 (  (None, 25, 25, 64)           192       ['conv2d_1041[0][0]']
 BatchNormalization)

 batch_normalization_1044 (  (None, 25, 25, 96)           288       ['conv2d_1044[0][0]']
 BatchNormalization)

 batch_normalization_1045 (  (None, 25, 25, 32)           96        ['conv2d_1045[0][0]']
 BatchNormalization)

 activation_1039 (Activatio  (None, 25, 25, 64)           0         ['batch_normalizatio
n_1039[0][
 n)                                                                 0]']

 activation_1041 (Activatio  (None, 25, 25, 64)           0         ['batch_normalizatio
n_1041[0][
 n)                                                                 0]']

 activation_1044 (Activatio  (None, 25, 25, 96)           0         ['batch_normalizatio
n_1044[0][
 n)                                                                 0]']

 activation_1045 (Activatio  (None, 25, 25, 32)           0         ['batch_normalizatio
n_1045[0][
 n)                                                                 0]']

 mixed0 (Concatenate)        (None, 25, 25, 256)          0         ['activation_1039[0]
```

```
 mixed0 (Concatenate)          (None, 25, 25, 256)         0          ['activation_1039[0]
[0]',
                                                                        'activation_1041[0
]	[0]',
                                                                        'activation_1044[0
][0]',
                                                                        'activation_1045[0
][0]']

 conv2d_1049 (Conv2D)          (None, 25, 25, 64)          16384      ['mixed0[0][0]']

 batch_normalization_1049 (    (None, 25, 25, 64)          192        ['conv2d_1049[0][0]']
 BatchNormalization)

 activation_1049 (Activatio    (None, 25, 25, 64)          0          ['batch_normalizatio
n_1049[0][
 n)                                                                   0]']

 conv2d_1047 (Conv2D)          (None, 25, 25, 48)          12288      ['mixed0[0][0]']

 conv2d_1050 (Conv2D)          (None, 25, 25, 96)          55296      ['activation_1049[0]
[0]']

 batch_normalization_1047 (    (None, 25, 25, 48)          144        ['conv2d_1047[0][0]']
 BatchNormalization)

 batch_normalization_1050 (    (None, 25, 25, 96)          288        ['conv2d_1050[0][0]']
 BatchNormalization)

 activation_1047 (Activatio    (None, 25, 25, 48)          0          ['batch_normalizatio
n_1047[0][
 n)                                                                   0]']

 activation_1050 (Activatio    (None, 25, 25, 96)          0          ['batch_normalizatio
n_1050[0][
 n)                                                                   0]']

 average_pooling2d_110 (Ave    (None, 25, 25, 256)         0          ['mixed0[0][0]']
 ragePooling2D)

 conv2d_1046 (Conv2D)          (None, 25, 25, 64)          16384      ['mixed0[0][0]']

 conv2d_1048 (Conv2D)          (None, 25, 25, 64)          76800      ['activation_1047[0]
[0]']

 conv2d_1051 (Conv2D)          (None, 25, 25, 96)          82944      ['activation_1050[0]
```

```
 conv2d_1051 (Conv2D)          (None, 25, 25, 96)          82944      ['activation_1050[0]
[0]']

 conv2d_1052 (Conv2D)          (None, 25, 25, 64)          16384      ['average_pooling2d_
110[0][0]'
                                                                       ]


 batch_normalization_1046 (    (None, 25, 25, 64)          192        ['conv2d_1046[0][0]']

 BatchNormalization)


 batch_normalization_1048 (    (None, 25, 25, 64)          192        ['conv2d_1048[0][0]']

 BatchNormalization)


 batch_normalization_1051 (    (None, 25, 25, 96)          288        ['conv2d_1051[0][0]']

 BatchNormalization)


 batch_normalization_1052 (    (None, 25, 25, 64)          192        ['conv2d_1052[0][0]']

 BatchNormalization)


 activation_1046 (Activatio    (None, 25, 25, 64)          0          ['batch_normalizatio
n_1046[0][
 n)                                                                    0]']


 activation_1048 (Activatio    (None, 25, 25, 64)          0          ['batch_normalizatio
n_1048[0][
 n)                                                                    0]']


 activation_1051 (Activatio    (None, 25, 25, 96)          0          ['batch_normalizatio
n_1051[0][
 n)                                                                    0]']


 activation_1052 (Activatio    (None, 25, 25, 64)          0          ['batch_normalizatio
n_1052[0][
 n)                                                                    0]']


 mixed1 (Concatenate)          (None, 25, 25, 288)         0          ['activation_1046[0]
[0]',
                                                                        'activation_1048[0
][0]',
                                                                        'activation_1051[0
][0]',
                                                                        'activation_1052[0
][0]']

 conv2d_1056 (Conv2D)          (None, 25, 25, 64)          18432      ['mixed1[0][0]']


 batch_normalization_1056 (    (None, 25, 25, 64)          192        ['conv2d_1056[0][0]']
```

```
 batch_normalization_1056 (  (None, 25, 25, 64)          192         ['conv2d_1056[0][0]']

 BatchNormalization)


 activation_1056 (Activatio  (None, 25, 25, 64)          0           ['batch_normalizatio
n_1056[0][
 n)                                                                  0]']


 conv2d_1054 (Conv2D)        (None, 25, 25, 48)          13824       ['mixed1[0][0]']


 conv2d_1057 (Conv2D)        (None, 25, 25, 96)          55296       ['activation_1056[0]
[0]']

 batch_normalization_1054 (  (None, 25, 25, 48)          144         ['conv2d_1054[0][0]']

 BatchNormalization)


 batch_normalization_1057 (  (None, 25, 25, 96)          288         ['conv2d_1057[0][0]']

 BatchNormalization)


 activation_1054 (Activatio  (None, 25, 25, 48)          0           ['batch_normalizatio
n_1054[0][
 n)                                                                  0]']


 activation_1057 (Activatio  (None, 25, 25, 96)          0           ['batch_normalizatio
n_1057[0][
 n)                                                                  0]']


 average_pooling2d_111 (Ave  (None, 25, 25, 288)         0           ['mixed1[0][0]']

 ragePooling2D)


 conv2d_1053 (Conv2D)        (None, 25, 25, 64)          18432       ['mixed1[0][0]']


 conv2d_1055 (Conv2D)        (None, 25, 25, 64)          76800       ['activation_1054[0]
[0]']

 conv2d_1058 (Conv2D)        (None, 25, 25, 96)          82944       ['activation_1057[0]
[0]']

 conv2d_1059 (Conv2D)        (None, 25, 25, 64)          18432       ['average_pooling2d_
111[0][0]'
                                                                     ]


 batch_normalization_1053 (  (None, 25, 25, 64)          192         ['conv2d_1053[0][0]']

 BatchNormalization)
```

```
 batch_normalization_1055 (  (None, 25, 25, 64)           192       ['conv2d_1055[0][0]']

 BatchNormalization)


 batch_normalization_1058 (  (None, 25, 25, 96)           288       ['conv2d_1058[0][0]']

 BatchNormalization)


 batch_normalization_1059 (  (None, 25, 25, 64)           192       ['conv2d_1059[0][0]']

 BatchNormalization)


 activation_1053 (Activatio  (None, 25, 25, 64)           0         ['batch_normalizatio
n_1053[0][
 n)                                                                 0]']


 activation_1055 (Activatio  (None, 25, 25, 64)           0         ['batch_normalizatio
n_1055[0][
 n)                                                                 0]']


 activation_1058 (Activatio  (None, 25, 25, 96)           0         ['batch_normalizatio
n_1058[0][
 n)                                                                 0]']


 activation_1059 (Activatio  (None, 25, 25, 64)           0         ['batch_normalizatio
n_1059[0][
 n)                                                                 0]']


 mixed2 (Concatenate)        (None, 25, 25, 288)          0         ['activation_1053[0]
[0]',
                                                                     'activation_1055[0
][0]',
                                                                     'activation_1058[0
][0]',
                                                                     'activation_1059[0
][0]']

 conv2d_1061 (Conv2D)        (None, 25, 25, 64)           18432     ['mixed2[0][0]']


 batch_normalization_1061 (  (None, 25, 25, 64)           192       ['conv2d_1061[0][0]']

 BatchNormalization)


 activation_1061 (Activatio  (None, 25, 25, 64)           0         ['batch_normalizatio
n_1061[0][
 n)                                                                 0]']


 conv2d_1062 (Conv2D)        (None, 25, 25, 96)           55296     ['activation_1061[0]
[0]']
```

```
 batch_normalization_1062 (  (None, 25, 25, 96)          288       ['conv2d_1062[0][0]']

 BatchNormalization)


 activation_1062 (Activatio  (None, 25, 25, 96)          0         ['batch_normalizatio
n_1062[0][
 n)                                                                0]']


 conv2d_1060 (Conv2D)        (None, 12, 12, 384)         995328    ['mixed2[0][0]']


 conv2d_1063 (Conv2D)        (None, 12, 12, 96)          82944     ['activation_1062[0]
[0]']

 batch_normalization_1060 (  (None, 12, 12, 384)         1152      ['conv2d_1060[0][0]']

 BatchNormalization)


 batch_normalization_1063 (  (None, 12, 12, 96)          288       ['conv2d_1063[0][0]']

 BatchNormalization)


 activation_1060 (Activatio  (None, 12, 12, 384)         0         ['batch_normalizatio
n_1060[0][
 n)                                                                0]']


 activation_1063 (Activatio  (None, 12, 12, 96)          0         ['batch_normalizatio
n_1063[0][
 n)                                                                0]']


 max_pooling2d_46 (MaxPooli  (None, 12, 12, 288)         0         ['mixed2[0][0]']

 ng2D)


 mixed3 (Concatenate)        (None, 12, 12, 768)         0         ['activation_1060[0]
[0]',
                                                                    'activation_1063[0
][0]',
                                                                    'max_pooling2d_46[
0][0]']

 conv2d_1068 (Conv2D)        (None, 12, 12, 128)         98304     ['mixed3[0][0]']


 batch_normalization_1068 (  (None, 12, 12, 128)         384       ['conv2d_1068[0][0]']

 BatchNormalization)


 activation_1068 (Activatio  (None, 12, 12, 128)         0         ['batch_normalizatio
n_1068[0][
 n)                                                                0]']
```

```
 n)                                                            0]']

 conv2d_1069 (Conv2D)        (None, 12, 12, 128)          114688    ['activation_1068[0]
[0]']

 batch_normalization_1069 (  (None, 12, 12, 128)          384       ['conv2d_1069[0][0]']

 BatchNormalization)

 activation_1069 (Activatio  (None, 12, 12, 128)          0         ['batch_normalizatio
n_1069[0][
 n)                                                                  0]']

 conv2d_1065 (Conv2D)        (None, 12, 12, 128)          98304     ['mixed3[0][0]']

 conv2d_1070 (Conv2D)        (None, 12, 12, 128)          114688    ['activation_1069[0]
[0]']

 batch_normalization_1065 (  (None, 12, 12, 128)          384       ['conv2d_1065[0][0]']

 BatchNormalization)

 batch_normalization_1070 (  (None, 12, 12, 128)          384       ['conv2d_1070[0][0]']

 BatchNormalization)

 activation_1065 (Activatio  (None, 12, 12, 128)          0         ['batch_normalizatio
n_1065[0][
 n)                                                                  0]']

 activation_1070 (Activatio  (None, 12, 12, 128)          0         ['batch_normalizatio
n_1070[0][
 n)                                                                  0]']

 conv2d_1066 (Conv2D)        (None, 12, 12, 128)          114688    ['activation_1065[0]
[0]']

 conv2d_1071 (Conv2D)        (None, 12, 12, 128)          114688    ['activation_1070[0]
[0]']

 batch_normalization_1066 (  (None, 12, 12, 128)          384       ['conv2d_1066[0][0]']

 BatchNormalization)

 batch_normalization_1071 (  (None, 12, 12, 128)          384       ['conv2d_1071[0][0]']

 BatchNormalization)

 activation_1066 (Activatio  (None, 12, 12, 128)          0         ['batch_normalizatio
```

```
 activation_1066 (Activatio  (None, 12, 12, 128)         0           ['batch_normalizatio
n_1066[0][
 n)                                                                  0]']

 activation_1071 (Activatio  (None, 12, 12, 128)         0           ['batch_normalizatio
n_1071[0][
 n)                                                                  0]']

 average_pooling2d_112 (Ave  (None, 12, 12, 768)         0           ['mixed3[0][0]']
 ragePooling2D)

 conv2d_1064 (Conv2D)        (None, 12, 12, 192)         147456      ['mixed3[0][0]']

 conv2d_1067 (Conv2D)        (None, 12, 12, 192)         172032      ['activation_1066[0]
[0]']

 conv2d_1072 (Conv2D)        (None, 12, 12, 192)         172032      ['activation_1071[0]
[0]']

 conv2d_1073 (Conv2D)        (None, 12, 12, 192)         147456      ['average_pooling2d_
112[0][0]'
                                                                     ]

 batch_normalization_1064 (  (None, 12, 12, 192)         576         ['conv2d_1064[0][0]']
 BatchNormalization)

 batch_normalization_1067 (  (None, 12, 12, 192)         576         ['conv2d_1067[0][0]']
 BatchNormalization)

 batch_normalization_1072 (  (None, 12, 12, 192)         576         ['conv2d_1072[0][0]']
 BatchNormalization)

 batch_normalization_1073 (  (None, 12, 12, 192)         576         ['conv2d_1073[0][0]']
 BatchNormalization)

 activation_1064 (Activatio  (None, 12, 12, 192)         0           ['batch_normalizatio
n_1064[0][
 n)                                                                  0]']

 activation_1067 (Activatio  (None, 12, 12, 192)         0           ['batch_normalizatio
n_1067[0][
 n)                                                                  0]']

 activation_1072 (Activatio  (None, 12, 12, 192)         0           ['batch_normalizatio
```

```
 activation_1072 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1072[0][
 n)                                                              0]']


 activation_1073 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1073[0][
 n)                                                              0]']


 mixed4 (Concatenate)        (None, 12, 12, 768)         0         ['activation_1064[0]
[0]',
                                                                  'activation_1067[0
][0]',
                                                                  'activation_1072[0
][0]',
                                                                  'activation_1073[0
][0]']

 conv2d_1078 (Conv2D)        (None, 12, 12, 160)         122880    ['mixed4[0][0]']


 batch_normalization_1078 (  (None, 12, 12, 160)         480       ['conv2d_1078[0][0]']

 BatchNormalization)


 activation_1078 (Activatio  (None, 12, 12, 160)         0         ['batch_normalizatio
n_1078[0][
 n)                                                              0]']


 conv2d_1079 (Conv2D)        (None, 12, 12, 160)         179200    ['activation_1078[0]
[0]']

 batch_normalization_1079 (  (None, 12, 12, 160)         480       ['conv2d_1079[0][0]']

 BatchNormalization)


 activation_1079 (Activatio  (None, 12, 12, 160)         0         ['batch_normalizatio
n_1079[0][
 n)                                                              0]']


 conv2d_1075 (Conv2D)        (None, 12, 12, 160)         122880    ['mixed4[0][0]']


 conv2d_1080 (Conv2D)        (None, 12, 12, 160)         179200    ['activation_1079[0]
[0]']

 batch_normalization_1075 (  (None, 12, 12, 160)         480       ['conv2d_1075[0][0]']

 BatchNormalization)


 batch_normalization_1080 (  (None, 12, 12, 160)         480       ['conv2d_1080[0][0]']

 BatchNormalization)
```

```
 activation_1075 (Activatio  (None, 12, 12, 160)          0         ['batch_normalizatio
n_1075[0][
 n)                                                                  0]']


 activation_1080 (Activatio  (None, 12, 12, 160)          0         ['batch_normalizatio
n_1080[0][
 n)                                                                  0]']


 conv2d_1076 (Conv2D)        (None, 12, 12, 160)          179200    ['activation_1075[0]
[0]']

 conv2d_1081 (Conv2D)        (None, 12, 12, 160)          179200    ['activation_1080[0]
[0]']

 batch_normalization_1076 (  (None, 12, 12, 160)          480       ['conv2d_1076[0][0]']

 BatchNormalization)


 batch_normalization_1081 (  (None, 12, 12, 160)          480       ['conv2d_1081[0][0]']

 BatchNormalization)


 activation_1076 (Activatio  (None, 12, 12, 160)          0         ['batch_normalizatio
n_1076[0][
 n)                                                                  0]']


 activation_1081 (Activatio  (None, 12, 12, 160)          0         ['batch_normalizatio
n_1081[0][
 n)                                                                  0]']


 average_pooling2d_113 (Ave  (None, 12, 12, 768)          0         ['mixed4[0][0]']

 ragePooling2D)


 conv2d_1074 (Conv2D)        (None, 12, 12, 192)          147456    ['mixed4[0][0]']


 conv2d_1077 (Conv2D)        (None, 12, 12, 192)          215040    ['activation_1076[0]
[0]']

 conv2d_1082 (Conv2D)        (None, 12, 12, 192)          215040    ['activation_1081[0]
[0]']

 conv2d_1083 (Conv2D)        (None, 12, 12, 192)          147456    ['average_pooling2d_
113[0][0]'
                                                                     ]


 batch_normalization_1074 (  (None, 12, 12, 192)          576       ['conv2d_1074[0][0]']

 BatchNormalization)
```

```
 BatchNormalization)

 batch_normalization_1077 (  (None, 12, 12, 192)         576       ['conv2d_1077[0][0]']

 BatchNormalization)

 batch_normalization_1082 (  (None, 12, 12, 192)         576       ['conv2d_1082[0][0]']

 BatchNormalization)

 batch_normalization_1083 (  (None, 12, 12, 192)         576       ['conv2d_1083[0][0]']

 BatchNormalization)

 activation_1074 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1074[0][
 n)                                                                 0]']

 activation_1077 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1077[0][
 n)                                                                 0]']

 activation_1082 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1082[0][
 n)                                                                 0]']

 activation_1083 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1083[0][
 n)                                                                 0]']

 mixed5 (Concatenate)        (None, 12, 12, 768)         0         ['activation_1074[0]
[0]',
                                                                     'activation_1077[0
][0]',
                                                                     'activation_1082[0
][0]',
                                                                     'activation_1083[0
][0]']

 conv2d_1088 (Conv2D)        (None, 12, 12, 160)         122880    ['mixed5[0][0]']

 batch_normalization_1088 (  (None, 12, 12, 160)         480       ['conv2d_1088[0][0]']

 BatchNormalization)

 activation_1088 (Activatio  (None, 12, 12, 160)         0         ['batch_normalizatio
n_1088[0][
 n)                                                                 0]']

 conv2d_1089 (Conv2D)        (None, 12, 12, 160)         179200    ['activation_1088[0]
```

```
 conv2d_1089 (Conv2D)         (None, 12, 12, 160)          179200    ['activation_1088[0]
[0]']

 batch_normalization_1089 (   (None, 12, 12, 160)          480       ['conv2d_1089[0][0]']

 BatchNormalization)


 activation_1089 (Activatio   (None, 12, 12, 160)          0         ['batch_normalizatio
n_1089[0][
 n)                                                                  0]']


 conv2d_1085 (Conv2D)         (None, 12, 12, 160)          122880    ['mixed5[0][0]']


 conv2d_1090 (Conv2D)         (None, 12, 12, 160)          179200    ['activation_1089[0]
[0]']

 batch_normalization_1085 (   (None, 12, 12, 160)          480       ['conv2d_1085[0][0]']

 BatchNormalization)


 batch_normalization_1090 (   (None, 12, 12, 160)          480       ['conv2d_1090[0][0]']

 BatchNormalization)


 activation_1085 (Activatio   (None, 12, 12, 160)          0         ['batch_normalizatio
n_1085[0][
 n)                                                                  0]']


 activation_1090 (Activatio   (None, 12, 12, 160)          0         ['batch_normalizatio
n_1090[0][
 n)                                                                  0]']


 conv2d_1086 (Conv2D)         (None, 12, 12, 160)          179200    ['activation_1085[0]
[0]']

 conv2d_1091 (Conv2D)         (None, 12, 12, 160)          179200    ['activation_1090[0]
[0]']

 batch_normalization_1086 (   (None, 12, 12, 160)          480       ['conv2d_1086[0][0]']

 BatchNormalization)


 batch_normalization_1091 (   (None, 12, 12, 160)          480       ['conv2d_1091[0][0]']

 BatchNormalization)


 activation_1086 (Activatio   (None, 12, 12, 160)          0         ['batch_normalizatio
n_1086[0][
 n)                                                                  0]']
```

```
 activation_1091 (Activatio  (None, 12, 12, 160)          0          ['batch_normalizatio
n_1091[0][
 n)                                                                  0]']


 average_pooling2d_114 (Ave  (None, 12, 12, 768)          0          ['mixed5[0][0]']

 ragePooling2D)


 conv2d_1084 (Conv2D)        (None, 12, 12, 192)          147456     ['mixed5[0][0]']


 conv2d_1087 (Conv2D)        (None, 12, 12, 192)          215040     ['activation_1086[0]
[0]']

 conv2d_1092 (Conv2D)        (None, 12, 12, 192)          215040     ['activation_1091[0]
[0]']

 conv2d_1093 (Conv2D)        (None, 12, 12, 192)          147456     ['average_pooling2d_
114[0][0]'
                                                                     ]


 batch_normalization_1084 (  (None, 12, 12, 192)          576        ['conv2d_1084[0][0]']

 BatchNormalization)


 batch_normalization_1087 (  (None, 12, 12, 192)          576        ['conv2d_1087[0][0]']

 BatchNormalization)


 batch_normalization_1092 (  (None, 12, 12, 192)          576        ['conv2d_1092[0][0]']

 BatchNormalization)


 batch_normalization_1093 (  (None, 12, 12, 192)          576        ['conv2d_1093[0][0]']

 BatchNormalization)


 activation_1084 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1084[0][
 n)                                                                  0]']


 activation_1087 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1087[0][
 n)                                                                  0]']


 activation_1092 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1092[0][
 n)                                                                  0]']
```

```
 activation_1093 (Activatio  (None, 12, 12, 192)          0         ['batch_normalizatio
n_1093[0][
 n)                                                                  0]']


 mixed6 (Concatenate)        (None, 12, 12, 768)          0         ['activation_1084[0]
[0]',
                                                                     'activation_1087[0
][0]',
                                                                     'activation_1092[0
][0]',
                                                                     'activation_1093[0
][0]']

 conv2d_1098 (Conv2D)        (None, 12, 12, 192)          147456    ['mixed6[0][0]']


 batch_normalization_1098 (  (None, 12, 12, 192)          576       ['conv2d_1098[0][0]']

 BatchNormalization)


 activation_1098 (Activatio  (None, 12, 12, 192)          0         ['batch_normalizatio
n_1098[0][
 n)                                                                  0]']


 conv2d_1099 (Conv2D)        (None, 12, 12, 192)          258048    ['activation_1098[0]
[0]']

 batch_normalization_1099 (  (None, 12, 12, 192)          576       ['conv2d_1099[0][0]']

 BatchNormalization)


 activation_1099 (Activatio  (None, 12, 12, 192)          0         ['batch_normalizatio
n_1099[0][
 n)                                                                  0]']


 conv2d_1095 (Conv2D)        (None, 12, 12, 192)          147456    ['mixed6[0][0]']


 conv2d_1100 (Conv2D)        (None, 12, 12, 192)          258048    ['activation_1099[0]
[0]']

 batch_normalization_1095 (  (None, 12, 12, 192)          576       ['conv2d_1095[0][0]']

 BatchNormalization)


 batch_normalization_1100 (  (None, 12, 12, 192)          576       ['conv2d_1100[0][0]']

 BatchNormalization)


 activation_1095 (Activatio  (None, 12, 12, 192)          0         ['batch_normalizatio
n_1095[0][
 n)                                                                  0]']
```

```
 n)                                                                0]']

 activation_1100 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1100[0][
 n)                                                                0]']

 conv2d_1096 (Conv2D)        (None, 12, 12, 192)          258048     ['activation_1095[0]
[0]']

 conv2d_1101 (Conv2D)        (None, 12, 12, 192)          258048     ['activation_1100[0]
[0]']

 batch_normalization_1096 (  (None, 12, 12, 192)          576        ['conv2d_1096[0][0]']
 BatchNormalization)

 batch_normalization_1101 (  (None, 12, 12, 192)          576        ['conv2d_1101[0][0]']
 BatchNormalization)

 activation_1096 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1096[0][
 n)                                                                0]']

 activation_1101 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1101[0][
 n)                                                                0]']

 average_pooling2d_115 (Ave  (None, 12, 12, 768)          0          ['mixed6[0][0]']
 ragePooling2D)

 conv2d_1094 (Conv2D)        (None, 12, 12, 192)          147456     ['mixed6[0][0]']

 conv2d_1097 (Conv2D)        (None, 12, 12, 192)          258048     ['activation_1096[0]
[0]']

 conv2d_1102 (Conv2D)        (None, 12, 12, 192)          258048     ['activation_1101[0]
[0]']

 conv2d_1103 (Conv2D)        (None, 12, 12, 192)          147456     ['average_pooling2d_
115[0][0]'
                                                                   ]

 batch_normalization_1094 (  (None, 12, 12, 192)          576        ['conv2d_1094[0][0]']
 BatchNormalization)

 batch_normalization_1097 (  (None, 12, 12, 192)          576        ['conv2d_1097[0][0]']
```

```
 batch_normalization_1097 (  (None, 12, 12, 192)         576       ['conv2d_1097[0][0]']
 BatchNormalization)


 batch_normalization_1102 (  (None, 12, 12, 192)         576       ['conv2d_1102[0][0]']
 BatchNormalization)


 batch_normalization_1103 (  (None, 12, 12, 192)         576       ['conv2d_1103[0][0]']
 BatchNormalization)


 activation_1094 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1094[0][
 n)                                                                 0]']


 activation_1097 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1097[0][
 n)                                                                 0]']


 activation_1102 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1102[0][
 n)                                                                 0]']


 activation_1103 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1103[0][
 n)                                                                 0]']


 mixed7 (Concatenate)        (None, 12, 12, 768)         0         ['activation_1094[0]
[0]',
                                                                     'activation_1097[0
][0]',
                                                                     'activation_1102[0
][0]',
                                                                     'activation_1103[0
][0]']

 conv2d_1106 (Conv2D)        (None, 12, 12, 192)         147456    ['mixed7[0][0]']


 batch_normalization_1106 (  (None, 12, 12, 192)         576       ['conv2d_1106[0][0]']
 BatchNormalization)


 activation_1106 (Activatio  (None, 12, 12, 192)         0         ['batch_normalizatio
n_1106[0][
 n)                                                                 0]']


 conv2d_1107 (Conv2D)        (None, 12, 12, 192)         258048    ['activation_1106[0]
[0]']


 batch_normalization_1107 (  (None, 12, 12, 192)         576       ['conv2d_1107[0][0]']
```

```
 batch_normalization_1107 (  (None, 12, 12, 192)          576        ['conv2d_1107[0][0]']
 BatchNormalization)


 activation_1107 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1107[0][
 n)                                                                  0]']


 conv2d_1104 (Conv2D)        (None, 12, 12, 192)          147456     ['mixed7[0][0]']


 conv2d_1108 (Conv2D)        (None, 12, 12, 192)          258048     ['activation_1107[0]
[0]']

 batch_normalization_1104 (  (None, 12, 12, 192)          576        ['conv2d_1104[0][0]']
 BatchNormalization)


 batch_normalization_1108 (  (None, 12, 12, 192)          576        ['conv2d_1108[0][0]']
 BatchNormalization)


 activation_1104 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1104[0][
 n)                                                                  0]']


 activation_1108 (Activatio  (None, 12, 12, 192)          0          ['batch_normalizatio
n_1108[0][
 n)                                                                  0]']


 conv2d_1105 (Conv2D)        (None, 5, 5, 320)            552960     ['activation_1104[0]
[0]']

 conv2d_1109 (Conv2D)        (None, 5, 5, 192)            331776     ['activation_1108[0]
[0]']

 batch_normalization_1105 (  (None, 5, 5, 320)            960        ['conv2d_1105[0][0]']
 BatchNormalization)


 batch_normalization_1109 (  (None, 5, 5, 192)            576        ['conv2d_1109[0][0]']
 BatchNormalization)


 activation_1105 (Activatio  (None, 5, 5, 320)            0          ['batch_normalizatio
n_1105[0][
 n)                                                                  0]']


 activation_1109 (Activatio  (None, 5, 5, 192)            0          ['batch_normalizatio
n_1109[0][
 n)                                                                  0]']
```

```
n)                                                                  0]']

 max_pooling2d_47 (MaxPooli  (None, 5, 5, 768)            0         ['mixed7[0][0]']

 ng2D)


 mixed8 (Concatenate)        (None, 5, 5, 1280)           0         ['activation_1105[0]
[0]',
                                                                     'activation_1109[0
][0]',
                                                                     'max_pooling2d_47[
0][0]']

 conv2d_1114 (Conv2D)        (None, 5, 5, 448)            573440    ['mixed8[0][0]']


 batch_normalization_1114 (  (None, 5, 5, 448)            1344      ['conv2d_1114[0][0]']

 BatchNormalization)


 activation_1114 (Activatio  (None, 5, 5, 448)            0         ['batch_normalizatio
n_1114[0][
 n)                                                                 0]']


 conv2d_1111 (Conv2D)        (None, 5, 5, 384)            491520    ['mixed8[0][0]']


 conv2d_1115 (Conv2D)        (None, 5, 5, 384)            1548288   ['activation_1114[0][
0]']

 batch_normalization_1111 (  (None, 5, 5, 384)            1152      ['conv2d_1111[0][0]']

 BatchNormalization)


 batch_normalization_1115 (  (None, 5, 5, 384)            1152      ['conv2d_1115[0][0]']

 BatchNormalization)


 activation_1111 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1111[0][
 n)                                                                 0]']


 activation_1115 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1115[0][
 n)                                                                 0]']


 conv2d_1112 (Conv2D)        (None, 5, 5, 384)            442368    ['activation_1111[0]
[0]']

 conv2d_1113 (Conv2D)        (None, 5, 5, 384)            442368    ['activation_1111[0]
[0]']
```

```
 conv2d_1116 (Conv2D)        (None, 5, 5, 384)            442368    ['activation_1115[0]
[0]']

 conv2d_1117 (Conv2D)        (None, 5, 5, 384)            442368    ['activation_1115[0]
[0]']

 average_pooling2d_116 (Ave  (None, 5, 5, 1280)           0         ['mixed8[0][0]']
 ragePooling2D)

 conv2d_1110 (Conv2D)        (None, 5, 5, 320)            409600    ['mixed8[0][0]']

 batch_normalization_1112 (  (None, 5, 5, 384)            1152      ['conv2d_1112[0][0]']
 BatchNormalization)

 batch_normalization_1113 (  (None, 5, 5, 384)            1152      ['conv2d_1113[0][0]']
 BatchNormalization)

 batch_normalization_1116 (  (None, 5, 5, 384)            1152      ['conv2d_1116[0][0]']
 BatchNormalization)

 batch_normalization_1117 (  (None, 5, 5, 384)            1152      ['conv2d_1117[0][0]']
 BatchNormalization)

 conv2d_1118 (Conv2D)        (None, 5, 5, 192)            245760    ['average_pooling2d_
116[0][0]'
                                                                    ]

 batch_normalization_1110 (  (None, 5, 5, 320)            960       ['conv2d_1110[0][0]']
 BatchNormalization)

 activation_1112 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1112[0][
 n)                                                                 0]']

 activation_1113 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1113[0][
 n)                                                                 0]']

 activation_1116 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1116[0][
 n)                                                                 0]']
```

```
 activation_1117 (Activatio  (None, 5, 5, 384)          0          ['batch_normalizatio
n_1117[0][
 n)                                                                0]']


 batch_normalization_1118 (  (None, 5, 5, 192)          576        ['conv2d_1118[0][0]']

 BatchNormalization)


 activation_1110 (Activatio  (None, 5, 5, 320)          0          ['batch_normalizatio
n_1110[0][
 n)                                                                0]']


 mixed9_0 (Concatenate)      (None, 5, 5, 768)          0          ['activation_1112[0]
[0]',
                                                                    'activation_1113[0
][0]']

 concatenate_22 (Concatenat  (None, 5, 5, 768)          0          ['activation_1116[0]
[0]',
 e)                                                                 'activation_1117[0
][0]']

 activation_1118 (Activatio  (None, 5, 5, 192)          0          ['batch_normalizatio
n_1118[0][
 n)                                                                0]']


 mixed9 (Concatenate)        (None, 5, 5, 2048)         0          ['activation_1110[0]
[0]',
                                                                    'mixed9_0[0][0]',
                                                                    'concatenate_22[0]
[0]',
                                                                    'activation_1118[0
][0]']

 conv2d_1123 (Conv2D)        (None, 5, 5, 448)          917504     ['mixed9[0][0]']


 batch_normalization_1123 (  (None, 5, 5, 448)          1344       ['conv2d_1123[0][0]']

 BatchNormalization)


 activation_1123 (Activatio  (None, 5, 5, 448)          0          ['batch_normalizatio
n_1123[0][
 n)                                                                0]']


 conv2d_1120 (Conv2D)        (None, 5, 5, 384)          786432     ['mixed9[0][0]']


 conv2d_1124 (Conv2D)        (None, 5, 5, 384)          1548288    ['activation_1123[0][
0]']

 batch_normalization_1120 (  (None, 5, 5, 384)          1152       ['conv2d_1120[0][0]']
```

```
 batch_normalization_1120 (  (None, 5, 5, 384)           1152      ['conv2d_1120[0][0]']
 BatchNormalization)


 batch_normalization_1124 (  (None, 5, 5, 384)           1152      ['conv2d_1124[0][0]']
 BatchNormalization)


 activation_1120 (Activatio  (None, 5, 5, 384)           0         ['batch_normalizatio
n_1120[0][
 n)                                                                0]']


 activation_1124 (Activatio  (None, 5, 5, 384)           0         ['batch_normalizatio
n_1124[0][
 n)                                                                0]']


 conv2d_1121 (Conv2D)        (None, 5, 5, 384)           442368    ['activation_1120[0]
[0]']

 conv2d_1122 (Conv2D)        (None, 5, 5, 384)           442368    ['activation_1120[0]
[0]']

 conv2d_1125 (Conv2D)        (None, 5, 5, 384)           442368    ['activation_1124[0]
[0]']

 conv2d_1126 (Conv2D)        (None, 5, 5, 384)           442368    ['activation_1124[0]
[0]']

 average_pooling2d_117 (Ave  (None, 5, 5, 2048)          0         ['mixed9[0][0]']
 ragePooling2D)


 conv2d_1119 (Conv2D)        (None, 5, 5, 320)           655360    ['mixed9[0][0]']


 batch_normalization_1121 (  (None, 5, 5, 384)           1152      ['conv2d_1121[0][0]']
 BatchNormalization)


 batch_normalization_1122 (  (None, 5, 5, 384)           1152      ['conv2d_1122[0][0]']
 BatchNormalization)


 batch_normalization_1125 (  (None, 5, 5, 384)           1152      ['conv2d_1125[0][0]']
 BatchNormalization)


 batch_normalization_1126 (  (None, 5, 5, 384)           1152      ['conv2d_1126[0][0]']
 BatchNormalization)
```

```
 conv2d_1127 (Conv2D)        (None, 5, 5, 192)            393216    ['average_pooling2d_
117[0][0]'
                                                                     ]


 batch_normalization_1119 (  (None, 5, 5, 320)            960       ['conv2d_1119[0][0]']

 BatchNormalization)


 activation_1121 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1121[0][
 n)                                                                  0]']


 activation_1122 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1122[0][
 n)                                                                  0]']


 activation_1125 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1125[0][
 n)                                                                  0]']


 activation_1126 (Activatio  (None, 5, 5, 384)            0         ['batch_normalizatio
n_1126[0][
 n)                                                                  0]']


 batch_normalization_1127 (  (None, 5, 5, 192)            576       ['conv2d_1127[0][0]']

 BatchNormalization)


 activation_1119 (Activatio  (None, 5, 5, 320)            0         ['batch_normalizatio
n_1119[0][
 n)                                                                  0]']


 mixed9_1 (Concatenate)      (None, 5, 5, 768)            0         ['activation_1121[0]
[0]',
                                                                      'activation_1122[0
][0]']

 concatenate_23 (Concatenat  (None, 5, 5, 768)            0         ['activation_1125[0]
[0]',
 e)                                                                   'activation_1126[0
][0]']

 activation_1127 (Activatio  (None, 5, 5, 192)            0         ['batch_normalizatio
n_1127[0][
 n)                                                                  0]']


 mixed10 (Concatenate)       (None, 5, 5, 2048)           0         ['activation_1119[0]
[0]',
                                                                      'mixed9_1[0][0]',

                                                                      'concatenate_23[0]
```

```
[0]',                                                           'activation_1127[0
][0]']

 average_pooling2d_118 (Ave  (None, 2, 2, 2048)          0         ['mixed10[0][0]']
 ragePooling2D)

 dropout_11 (Dropout)        (None, 2, 2, 2048)          0         ['average_pooling2d_
118[0][0]'
                                                                   ]

 flatten_12 (Flatten)        (None, 8192)                0         ['dropout_11[0][0]']

 dense_12 (Dense)            (None, 101)                 827493    ['flatten_12[0][0]']

==========================================================================================
Total params: 22630277 (86.33 MB)
Trainable params: 22595845 (86.20 MB)
Non-trainable params: 34432 (134.50 KB)
__________________________________________________________________________________________
```

In [ ]:

```
opt = SGD(lr=.1, momentum=.9)
model3.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
```

In [ ]:

```
from keras.models import load_model
model3 = load_model("/kaggle/input/food-101-model/tensorflow2/food-101/1/model_food_101.h
5")
```

In [ ]:

```
from tensorflow.keras.callbacks import EarlyStopping
results3 = model3.fit(train_data, epochs=10, validation_data=test_data,
                  steps_per_epoch=len(train_data), validation_steps=len(test_data),
                  callbacks = EarlyStopping(patience=2, monitor='val_accuracy',restore_b
est_weights=True)
                  )
```

```
Epoch 1/10
```

```
2024-03-14 23:31:53.916425: E tensorflow/core/grappler/optimizers/meta_optimizer.cc:961]
layout failed: INVALID_ARGUMENT: Size of values 0 does not match size of permutation 4 @
fanin shape inmodel_1/dropout_11/dropout/SelectV2-2-TransposeNHWCToNCHW-LayoutOptimizer
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1710459118.259536     186 device_compiler.h:186] Compiled cluster using XLA!
This line is logged at most once for the lifetime of the process.
```

```
909/909 [==============================] - 1009s 1s/step - loss: 0.0816 - accuracy: 0.992
4 - val_loss: 1.6280 - val_accuracy: 0.7065
Epoch 2/10
909/909 [==============================] - 829s 912ms/step - loss: 0.0646 - accuracy: 0.9
962 - val_loss: 1.4801 - val_accuracy: 0.7417
Epoch 3/10
909/909 [==============================] - 830s 913ms/step - loss: 0.0562 - accuracy: 0.9
970 - val_loss: 1.4059 - val_accuracy: 0.7487
Epoch 4/10
909/909 [==============================] - 829s 911ms/step - loss: 0.0460 - accuracy: 0.9
```

```
986 - val_loss: 1.3613 - val_accuracy: 0.7535
Epoch 5/10
909/909 [==============================] - 828s 911ms/step - loss: 0.0379 - accuracy: 0.9
995 - val_loss: 1.2399 - val_accuracy: 0.7697
Epoch 6/10
909/909 [==============================] - 828s 911ms/step - loss: 0.0325 - accuracy: 0.9
997 - val_loss: 1.2145 - val_accuracy: 0.7693
Epoch 7/10
909/909 [==============================] - 828s 911ms/step - loss: 0.0284 - accuracy: 0.9
997 - val_loss: 1.1774 - val_accuracy: 0.7716
Epoch 8/10
909/909 [==============================] - 828s 911ms/step - loss: 0.0245 - accuracy: 0.9
999 - val_loss: 1.1357 - val_accuracy: 0.7722
Epoch 9/10
909/909 [==============================] - 828s 910ms/step - loss: 0.0216 - accuracy: 0.9
999 - val_loss: 1.1133 - val_accuracy: 0.7721
Epoch 10/10
909/909 [==============================] - 827s 910ms/step - loss: 0.0191 - accuracy: 1.0
000 - val_loss: 1.0886 - val_accuracy: 0.7736
```

In [ ]:

```
loss, acc = model3.evaluate(test_data)
```

```
101/101 [==============================] - 41s 402ms/step - loss: 1.0886 - accuracy: 0.77
36
```

In [ ]:

```
print("Test Accuracy:",round(acc*100,2),"%","\nTest Loss:",loss)
```

```
Test Accuracy: 77.36 %
Test Loss: 1.0886189937591553
```

In [ ]:

```
model = model3
```

Training again

In [ ]:

```
from tensorflow.keras.callbacks import EarlyStopping
results3 = model3.fit(train_data, epochs=10, validation_data=test_data,
                   steps_per_epoch=len(train_data), validation_steps=len(test_data),
                   callbacks = EarlyStopping(patience=2, monitor='val_accuracy',restore_b
est_weights=True)
                   )
```

```
Epoch 1/10
909/909 [==============================] - 829s 912ms/step - loss: 0.0173 - accuracy: 0.9
999 - val_loss: 1.0805 - val_accuracy: 0.7749
Epoch 2/10
909/909 [==============================] - 828s 911ms/step - loss: 0.0158 - accuracy: 1.0
000 - val_loss: 1.0645 - val_accuracy: 0.7743
Epoch 3/10
909/909 [==============================] - 828s 910ms/step - loss: 0.0145 - accuracy: 1.0
000 - val_loss: 1.0611 - val_accuracy: 0.7733
```

In [ ]:

```
loss, acc = model3.evaluate(test_data)
```

```
101/101 [==============================] - 41s 406ms/step - loss: 1.0805 - accuracy: 0.77
49
```

In [ ]:

```
print("Test Accuracy:",round(acc*100,2),"%","\nTest Loss:",round(loss,4))
```

```
Test Accuracy: 77.49 %
```

```
Test Loss: 1.0805
```

In [ ]:

```
results3 = {"accuracy":    [0.9924,0.9962,0.9970,0.9986,0.9995,0.9997,0.9997,0.9999,0.99
99,1.0000,0.9999,1.0000,1.0000],
            "loss":        [0.0816,0.0646,0.0562,0.0460,0.0379,0.0325,0.0284,0.0245,0.02
16,0.0191,0.0173,0.0158,0.0145],
            "val_accuracy":[0.7065,0.7417,0.7487,0.7535,0.7697,0.7693,0.7716,0.7722,0.77
21,0.7736,0.7749,0.7743,0.7733],
            "val_loss":    [1.6280,1.4801,1.4059,1.3613,1.2399,1.2145,1.1774,1.1357,1.11
33,1.0886,1.0805,1.0645,1.0611]}
```

In [ ]:

```
model3.save("model_food_1012.h5")
```

```
/opt/conda/lib/python3.10/site-packages/keras/src/engine/training.py:3103: UserWarning: Y
ou are saving your model as an HDF5 file via `model.save()`. This file format is consider
ed legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model
.keras')`.
  saving_api.save_model(
```

In [ ]:

```
results3['val_accuracy']
```

Out[ ]:

```
[0.7065,
 0.7417,
 0.7487,
 0.7535,
 0.7697,
 0.7693,
 0.7716,
 0.7722,
 0.7721,
 0.7736,
 0.7749,
 0.7743,
 0.7733]
```

In [ ]:

```
results3['val_loss']
```

Out[ ]:

```
[1.628,
 1.4801,
 1.4059,
 1.3613,
 1.2399,
 1.2145,
 1.1774,
 1.1357,
 1.1133,
 1.0886,
 1.0805,
 1.0645,
 1.0611]
```

In [ ]:

```
i = results3['val_loss'].index(round(loss,4))
i
```

Out[ ]:

```
10
```

In [ ]:

```
results3['val_accuracy'][i]
```

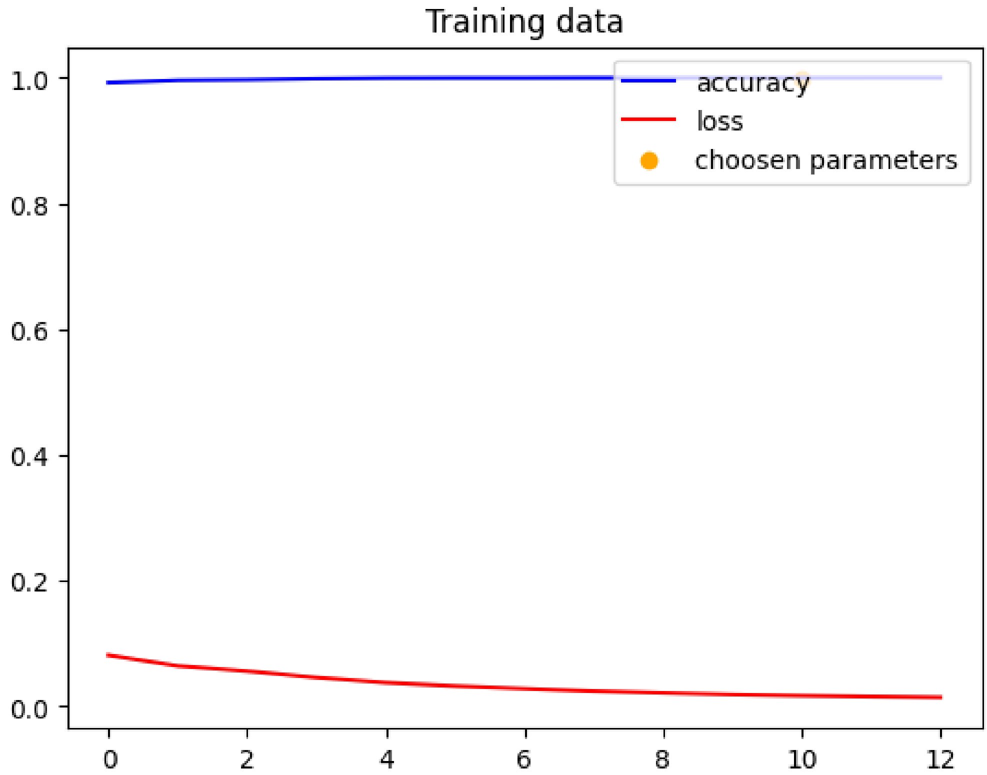
Out[ ]:

```
0.7749
```

Visualize training history

In [ ]:

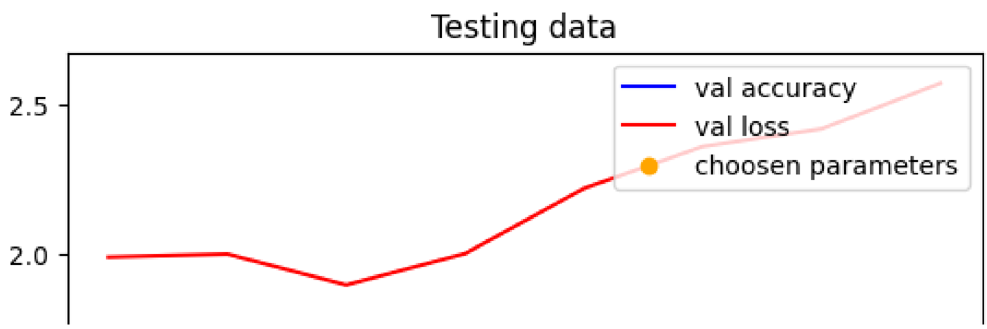
```
fig = plt.figure()
plt.plot(results3['accuracy'], c='blue', label='accuracy')
plt.plot(results3['loss'], c='red', label='loss')
plt.scatter(i,results3['accuracy'][i], c='orange', marker='o', label='choosen parameters
')
plt.title('Training data')
plt.legend(loc='upper right')
plt.show()
```

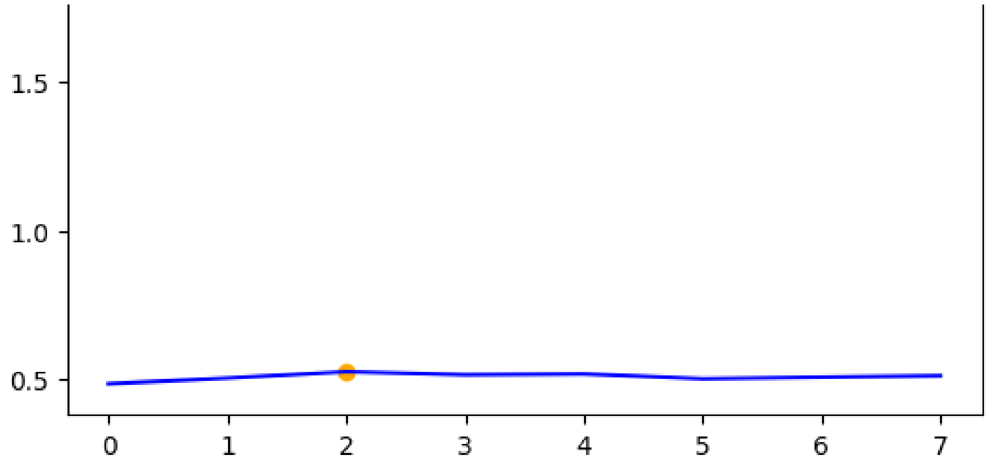


Visualize testing history

In [ ]:

```
fig = plt.figure()
plt.plot(results3.history['val_accuracy'], c='blue', label='val accuracy')
plt.plot(results3.history['val_loss'], c='red', label='val loss')
plt.scatter(i,results3.history['val_accuracy'][i], c='orange', marker='o', label='choose
n parameters')
plt.title('Testing data')
plt.legend(loc='upper right')
plt.show()
```





In [ ]:

```
yp = model3.predict(test_data).argmax(axis=1).reshape(-1,)
yp.shape
```

```
101/101 [==============================] - 48s 468ms/step
```

Out[ ]:

```
(10100,)
```

In [ ]:

```
m = pd.crosstab(test_data.labels, yp, rownames=['Actual'], colnames=['Predicted'])
m
```

Out[ ]:

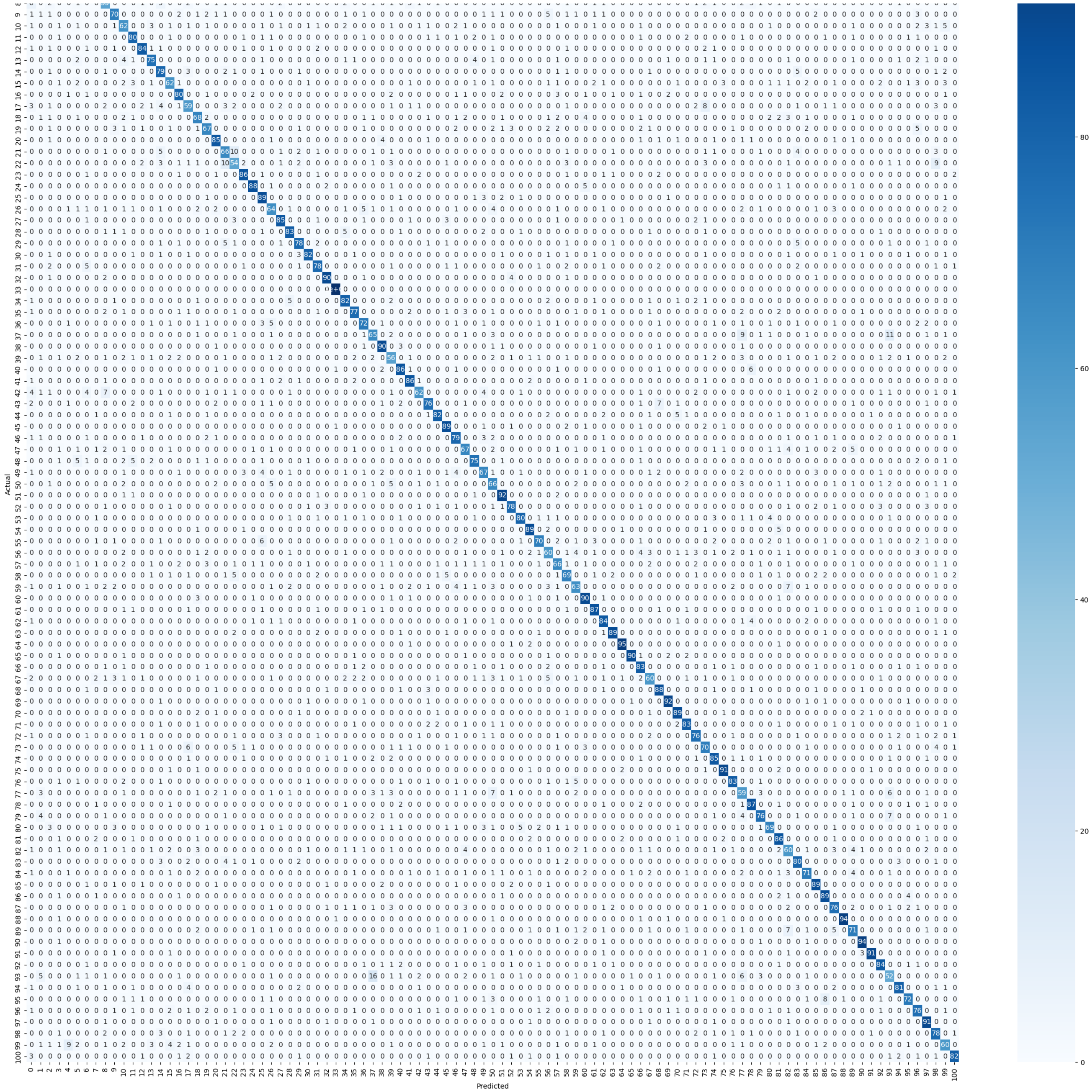
Predicted	0	1	2	3	4	5	6	7	8	9	...	91	92	93	94	95	96	97	98	99	100
Actual																					
0	56	1	1	0	0	0	0	0	8	1	...	0	0	1	1	0	0	0	1	1	2
1	0	77	0	1	0	0	0	0	0	0	...	0	0	5	0	0	0	0	0	0	0
2	3	0	82	0	0	0	0	0	1	0	...	0	2	0	0	1	0	0	0	1	0
3	0	0	0	80	1	1	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
4	0	0	0	1	72	1	0	0	0	0	...	0	0	1	0	0	1	0	0	8	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
96	1	0	0	0	0	0	0	0	0	1	...	0	1	0	0	0	76	0	0	1	0
97	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	91	0	0	0
98	0	0	0	1	0	0	0	0	2	0	...	1	0	0	0	0	0	0	78	0	1
99	0	1	1	1	9	2	0	0	1	0	...	0	0	0	0	1	0	0	0	60	0
100	3	0	0	0	0	0	0	0	0	0	...	0	0	1	2	0	1	0	1	0	82

101 rows × 101 columns

In [ ]:

```
print("Heatmap\n")
plt.figure(figsize=(30, 30))
sn.heatmap(m,annot=True, cmap='Blues')
plt.show()
```

```
Heatmap
```



Testing Samples

In [ ]:

```
print(os.listdir("/kaggle/input/food-101/food-101/food-101/images/"))
```

```
['macarons', 'french_toast', 'lobster_bisque', 'prime_rib', 'pork_chop', 'guacamole', 'ba
by_back_ribs', 'mussels', 'beef_carpaccio', 'poutine', 'hot_and_sour_soup', 'seaweed_sala
d', 'foie_gras', 'dumplings', 'peking_duck', 'takoyaki', 'bibimbap', 'falafel', 'pulled_p
ork_sandwich', 'lobster_roll_sandwich', 'carrot_cake', 'beet_salad', 'panna_cotta', 'donu
ts', 'red_velvet_cake', 'grilled_cheese_sandwich', 'cannoli', 'spring_rolls', 'shrimp_and
_grits', 'clam_chowder', 'omelette', 'fried_calamari', 'caprese_salad', 'oysters', 'scall
ops', 'ramen', 'grilled_salmon', 'croque_madame', 'filet_mignon', 'hamburger', 'spaghetti
_carbonara', 'miso_soup', 'bread_pudding', 'lasagna', 'crab_cakes', 'cheesecake', 'spaghe
tti_bolognese', 'cup_cakes', 'creme_brulee', 'waffles', 'fish_and_chips', 'paella', 'maca
roni_and_cheese', 'chocolate_mousse', 'ravioli', 'chicken_curry', 'caesar_salad', 'nachos
', 'tiramisu', 'frozen_yogurt', 'ice_cream', 'risotto', 'club_sandwich', 'strawberry_shor
tcake', 'steak', 'churros', 'garlic_bread', 'baklava', 'bruschetta', 'hummus', 'chicken_w
ings', 'greek_salad', 'tuna_tartare', 'chocolate_cake', 'gyoza', 'eggs_benedict', 'devile
d_eggs', 'samosa', 'sushi', 'breakfast_burrito', 'ceviche', 'beef_tartare', 'apple_pie',
'.DS_Store', 'huevos_rancheros', 'beignets', 'pizza', 'edamame', 'french_onion_soup', 'ho
t_dog', 'tacos', 'chicken_quesadilla', 'pho', 'gnocchi', 'pancakes', 'fried_rice', 'chees
e_plate', 'onion_rings', 'escargots', 'sashimi', 'pad_thai', 'french_fries']
```

In [ ]:

```
print("Macarons Sample")
macarons = load_img("/kaggle/input/food-101/food-101/food-101/images/macarons/2428554.jpg
",target_size=(224,224))
macarons
```

Macarons Sample

Out[ ]:

In [ ]:

```
print("Pizza Sample")
pizza = load_img("/kaggle/input/food-101/food-101/food-101/images/pizza/768276.jpg",targe
t_size=(224,224,3))
pizza
```

Pizza Sample

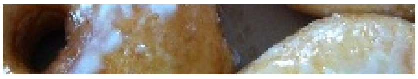
Out[ ]:

In [ ]:

```
print("Donuts Sample")
donuts = load_img("/kaggle/input/food-101/food-101/food-101/images/donuts/2563686.jpg",ta
rget_size=(224,224,3))
donuts
```

Donuts Sample

Out[ ]:



In [ ]:

```
print("Frensh Toast Sample")
toast = load_img("/kaggle/input/food-101/food-101/food-101/images/french_toast/2769309.jpg",target_size=(224,224,3))
toast
```

```
Frensh Toast Sample
```

Out[ ]:

In [ ]:

```
print("French_fries Sample")
fries = load_img("/kaggle/input/food-101/food-101/food-101/images/french_fries/2246621.jpg",target_size=(224,224))
fries
```

```
French_fries Sample
```

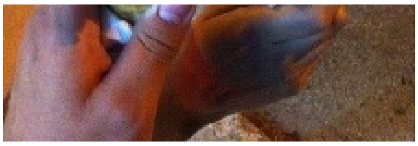
Out[ ]:

In [ ]:

```
print("Ice Cream Sample")
ice = load_img("/kaggle/input/food-101/food-101/food-101/images/ice_cream/579407.jpg",target_size=(224,224))
ice
```

```
Ice Cream Sample
```

Out[ ]:



In [ ]:

```
fig = plt.figure(figsize=(14,7))
# plt.grid=False
fig.add_subplot(1, 6, 1)
plt.axis('off')
plt.imshow(macarons)
plt.title("Macarons")
# _____________________________________________
fig.add_subplot(1, 6, 2)
plt.axis('off')
plt.imshow(fries)
plt.title("Frensh Fries")
# _____________________________________________
fig.add_subplot(1, 6, 3)
plt.axis('off')
plt.imshow(ice)
plt.title("Ice Cream")
# _____________________________________________
fig.add_subplot(1, 6, 4)
plt.axis('off')
plt.imshow(toast)
plt.title("Frensh Toast")
# _____________________________________________
fig.add_subplot(1, 6, 5)
plt.axis('off')
plt.imshow(pizza)
plt.title("Pizza")
# _____________________________________________
fig.add_subplot(1, 6, 6)
plt.axis('off')
plt.imshow(donuts)
plt.title("Donuts")
```

Out[ ]:

```
Text(0.5, 1.0, 'Donuts')
```

In [ ]:

```
macarons = img_to_array(macarons)
fries = img_to_array(fries)
ice = img_to_array(ice)
pizza = img_to_array(pizza)
donuts = img_to_array(donuts)
toast = img_to_array(toast)
macarons = macarons/255
fries = fries/255
ice = ice/255
pizza = pizza/255
donuts = donuts/255
toast = toast/255
macarons = macarons.reshape(1,224,224,3)
fries = fries.reshape(1,224,224,3)
ice = ice.reshape(1,224,224,3)
pizza = pizza.reshape(1,224,224,3)
donuts = donuts.reshape(1,224,224,3)
```

```
toast = toast.reshape(1,224,224,3)

macarons.shape
```

Out[ ]:

```
(1, 224, 224, 3)
```

Samples Predicting

In [ ]:

```
p1 = (model.predict(macarons)).argmax()

print("Class ",p1,": ",values[p1],sep='')
print(calories[p1],'\nNote:',s)
```

```
1/1 [==============================] - 3s 3s/step
Class 63: macarons
Macarons: ~4 calories per gram
Note: These values are approximations and can vary based on factors such as ingredients a
nd cooking methods.
```

In [ ]:

```
p2 = (model.predict(fries)).argmax()

print("Class ",p2,": ",values[p2],sep='')
print(calories[p2],'\nNote:',s)
```

```
1/1 [==============================] - 0s 28ms/step
Class 40: french_fries
French Fries: ~3.5 calories per gram
Note: These values are approximations and can vary based on factors such as ingredients a
nd cooking methods.
```

In [ ]:

```
p3 = (model.predict(ice)).argmax()

print("Class ",p3,": ",values[p3],sep='')
print(calories[p3],'\nNote:',s)
```

```
1/1 [==============================] - 0s 24ms/step
Class 58: ice_cream
Ice Cream: ~2 calories per gram
Note: These values are approximations and can vary based on factors such as ingredients a
nd cooking methods.
```

In [ ]:

```
p4 = (model.predict(pizza)).argmax()

print("Class ",p4,": ",values[p4],sep='')
print(calories[p4],'\nNote:',s)
```

```
1/1 [==============================] - 0s 29ms/step
Class 76: pizza
Pizza: ~2.5 calories per gram
Note: These values are approximations and can vary based on factors such as ingredients a
nd cooking methods.
```

In [ ]:

```
p5 = (model.predict(donuts)).argmax()

print("Class ",p5,": ",values[p5],sep='')
print(calories[p5],'\nNote:',s)
```

```
1/1 [==============================] - 0s 28ms/step
Class 31: donuts
```

```
Donuts: ~4 calories per gram
Note: These values are approximations and can vary based on factors such as ingredients a
nd cooking methods.
```

In [ ]:

```
p6 = (model.predict(toast)).argmax()

print("Class ",p6,": ",values[p6],sep='')
print(calories[p6],'\nNote:',s)
```

```
1/1 [==============================] - 0s 28ms/step
Class 42: french_toast
French Toast: ~2 calories per gram
Note: These values are approximations and can vary based on factors such as ingredients a
nd cooking methods.
```