

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
```

```
model_res=tf.keras.applications.resnet50.ResNet50()
print(model_res.input_shape)
model_res.summary()
```

Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50\\_weights\\_tf\\_dim\\_ordering\\_t102967424/102967424](https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_t102967424/102967424) [=====] - 1s 0us/step  
(None, 224, 224, 3)  
Model: "resnet50"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	[]
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	['input_1[0][0]']
conv1_conv (Conv2D)	(None, 112, 112, 64)	9472	['conv1_pad[0][0]']
conv1_bn (BatchNormalization)	(None, 112, 112, 64)	256	['conv1_conv[0][0]']
conv1_relu (Activation)	(None, 112, 112, 64)	0	['conv1_bn[0][0]']
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64)	0	['conv1_relu[0][0]']
pool1_pool (MaxPooling2D)	(None, 56, 56, 64)	0	['pool1_pad[0][0]']
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 64)	4160	['pool1_pool[0][0]']
conv2_block1_1_bn (BatchNormalization)	(None, 56, 56, 64)	256	['conv2_block1_1_conv[0][0]']
conv2_block1_1_relu (Activation)	(None, 56, 56, 64)	0	['conv2_block1_1_bn[0][0]']
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 64)	36928	['conv2_block1_1_relu[0][0]']
conv2_block1_2_bn (BatchNormalization)	(None, 56, 56, 64)	256	['conv2_block1_2_conv[0][0]']
conv2_block1_2_relu (Activation)	(None, 56, 56, 64)	0	['conv2_block1_2_bn[0][0]']
conv2_block1_0_conv (Conv2D)	(None, 56, 56, 256)	16640	['pool1_pool[0][0]']
conv2_block1_3_conv (Conv2D)	(None, 56, 56, 256)	16640	['conv2_block1_2_relu[0][0]']
conv2_block1_0_bn (BatchNormalization)	(None, 56, 56, 256)	1024	['conv2_block1_0_conv[0][0]']
conv2_block1_3_bn (BatchNormalization)	(None, 56, 56, 256)	1024	['conv2_block1_3_conv[0][0]']
conv2_block1_add (Add)	(None, 56, 56, 256)	0	['conv2_block1_0_bn[0][0]', 'conv2_block1_3_bn[0][0]']
conv2_block1_out (Activation)	(None, 56, 56, 256)	0	['conv2_block1_add[0][0]']

```
model = tf.keras.applications.mobilenet_v2.MobileNetV2()
print(model.input_shape)
model.summary()
```

Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/mobilenet\\_v2/mobilenet\\_v2\\_weights\\_tf\\_dim\\_ordering\\_tf\\_data\\_format/14536120/14536120](https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_data_format/14536120/14536120) [=====] - 0s 0us/step  
(None, 224, 224, 3)  
Model: "mobilenetv2\_1.00\_224"

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	[]
Conv1 (Conv2D)	(None, 112, 112, 32)	864	['input_2[0][0]']
bn_Conv1 (BatchNormalization)	(None, 112, 112, 32)	128	['Conv1[0][0]']

on)

Conv1_relu (ReLU)	(None, 112, 112, 32)	0	['bn_Conv1[0][0]']
expanded_conv_depthwise (DepthwiseConv2D)	(None, 112, 112, 32)	288	['Conv1_relu[0][0]']
expanded_conv_depthwise_BN (BatchNormalization)	(None, 112, 112, 32)	128	['expanded_conv_depthwise[0][0]']
expanded_conv_depthwise_relu (ReLU)	(None, 112, 112, 32)	0	['expanded_conv_depthwise_BN[0][0]']
expanded_conv_project (Conv2D)	(None, 112, 112, 16)	512	['expanded_conv_depthwise_relu[0][0]']
expanded_conv_project_BN (BatchNormalization)	(None, 112, 112, 16)	64	['expanded_conv_project[0][0]']
block_1_expand (Conv2D)	(None, 112, 112, 96)	1536	['expanded_conv_project_BN[0][0]']
block_1_expand_BN (BatchNormalization)	(None, 112, 112, 96)	384	['block_1_expand[0][0]']
block_1_expand_relu (ReLU)	(None, 112, 112, 96)	0	['block_1_expand_BN[0][0]']
block_1_pad (ZeroPadding2D)	(None, 113, 113, 96)	0	['block_1_expand_relu[0][0]']
block_1_depthwise (DepthwiseConv2D)	(None, 56, 56, 96)	864	['block_1_pad[0][0]']
block_1_depthwise_BN (BatchNormalization)	(None, 56, 56, 96)	384	['block_1_depthwise[0][0]']
block_1_depthwise_relu (ReLU)	(None, 56, 56, 96)	0	['block_1_depthwise_BN[0][0]']
block_1_project (Conv2D)	(None, 56, 56, 24)	2304	['block_1_depthwise_relu[0][0]']
block_1_project_BN (BatchNormalization)	(None, 56, 56, 24)	96	['block_1_project[0][0]']

```
##### SINGLE IMAGE #####
# Read the image using TensorFlow.
tf_image = tf.io.read_file('/content/drive/MyDrive/ALL_DS/roses/10503217854_e66a804309.jpg')
decoded_image = tf.image.decode_image(tf_image)
print(decoded_image.shape)
plt.imshow(decoded_image)
plt.axis('off')
##### Load Model
model = tf.keras.applications.mobilenet_v2.MobileNetV2()
print(model.input_shape)
##### Resize and add batch
image_resized = tf.image.resize(decoded_image, (224, 224))
print(image_resized.shape)
image_batch = tf.expand_dims(image_resized, axis=0)
print(image_batch.shape)
##### TL
image_batch_pre = tf.keras.applications.mobilenet_v2.preprocess_input(image_batch)
# Forward pass through the model to make predictions.
preds = model.predict(image_batch_pre)
print(preds.shape)
#print(preds)
```

```
(333, 500, 3)
(None, 224, 224, 3)
(224, 224, 3)
(1, 224, 224, 3)
1/1 [=====] - 1s 1s/step
(1, 1000)
```



```
decoded_preds = tf.keras.applications.mobilenet_v2.decode_predictions(preds=preds,top=5)
print(decoded_preds)
print(decoded_preds[0][0])
for i in range(5):
    print('Predicted class',decoded_preds[0][i][1], 'Predcited Probability ',decoded_preds[0][i][2])
```

```
↳ Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet\_class\_index.json
35363/35363 [=====] - 0s 0us/step
[(['n03724870', 'mask', 0.43171886), ('n04447861', 'toilet_seat', 0.023408927), ('n04398044', 'teapot', 0.022900498), ('n045221
('n03724870', 'mask', 0.43171886)
Predicted class mask Predcited Probability 0.43171886
Predicted class toilet_seat Predcited Probability 0.023408927
Predicted class teapot Predcited Probability 0.022900498
Predicted class vase Predcited Probability 0.012807392
Predicted class theater_curtain Predcited Probability 0.012717216
```

```
#print(preds)
```

```
[[2.07189692e-06 7.21743390e-07 2.97598490e-07 8.08408700e-08
1.86580030e-06 3.15135907e-07 1.61211133e-07 3.42671337e-05
1.28105254e-04 1.03630218e-05 4.00443651e-06 1.00267380e-05
1.56725400e-06 4.65759513e-06 4.20681545e-06 9.40726386e-06
6.63275750e-06 3.53526302e-05 6.60053774e-06 5.99054010e-06
3.92190896e-06 4.57718343e-05 2.74253307e-06 3.54458083e-04
1.10896581e-05 7.59649174e-06 1.74668332e-06 1.79749122e-06
3.66013865e-06 2.36232950e-06 1.42566387e-06 8.46480634e-06
3.58730540e-06 9.47898968e-07 2.19353046e-06 1.04163234e-06
5.65429627e-06 1.13916246e-06 1.82506801e-06 7.19807394e-06
7.74973250e-06 7.51871198e-07 1.96546375e-06 1.39398310e-06
1.31737727e-06 1.61468779e-06 6.75377942e-06 2.64600885e-06
4.86583167e-06 2.31606023e-07 5.43163708e-07 1.89065304e-06
1.89673563e-06 5.26753456e-06 7.14778480e-07 3.06043898e-06
1.43367674e-06 4.00558804e-07 8.55583096e-07 1.58352634e-06
4.37677670e-07 7.38059668e-07 4.58133599e-07 4.50449943e-06
1.35713969e-06 1.57036743e-06 1.56419375e-07 2.44903873e-07
1.35445987e-07 3.29840304e-07 2.11212154e-07 8.58414978e-07
1.59658236e-06 5.62686409e-07 5.85359714e-07 6.62739239e-07
3.05580613e-07 7.42866519e-07 6.78770164e-07 1.81874350e-06
4.82459200e-06 8.55877261e-06 4.01084071e-05 9.95528626e-06
1.80182215e-05 1.25787219e-05 1.04856634e-04 7.90937265e-05
6.36480627e-06 7.19515083e-04 1.95389907e-06 7.99649399e-07
1.19953825e-06 8.75910189e-07 2.22766685e-05 3.18729292e-07
3.16192700e-06 4.12625741e-05 3.25097994e-06 4.35423106e-04
1.73759108e-05 4.16741159e-06 2.52071800e-06 3.82425378e-06
9.40124664e-05 3.16369142e-06 4.92521963e-07 3.70953444e-07
1.12310795e-06 1.10754047e-06 1.14031241e-06 7.91202694e-07
9.45295369e-07 1.19931465e-05 4.83991471e-06 4.67449627e-06
5.00382100e-07 1.92859829e-07 3.02364015e-07 5.37953952e-07
1.22546794e-06 4.09158162e-07 1.60455414e-07 1.07876922e-07
1.28930844e-06 2.17798242e-06 1.66952998e-06 2.84796606e-05
3.32325703e-06 3.47128189e-05 2.22692643e-06 1.17736738e-06
5.31547448e-06 5.16659441e-07 2.21272057e-05 1.55189150e-06
1.08952618e-05 1.60212148e-05 2.87346484e-06 9.95481241e-06
1.63845664e-06 2.21921573e-06 1.20353855e-06 1.15771709e-05
1.45594777e-05 1.60043564e-04 3.13840545e-04 2.34672243e-07
1.63138353e-07 1.28452393e-05 2.08370002e-05 2.01905309e-03
```

2.22187908e-03 1.07350654e-03 2.96242326e-03 1.14957790e-03  
3.26517806e-03 1.78894657e-03 4.34945367e-04 1.09309133e-03  
3.98897333e-03 5.41598827e-04 5.56588359e-03 1.22504157e-03  
6.40369952e-04 1.02948034e-04 1.02863531e-03 1.55165256e-03  
1.50354905e-03 2.11785012e-03 1.11970119e-03 3.60211037e-04  
1.30466383e-03 2.29835656e-04 1.82493671e-03 2.07524491e-03  
1.40143111e-02 1.68332816e-04 9.05286288e-04 6.45949389e-04  
9.69798130e-04 3.11168320e-02 6.33299176e-04 4.09693544e-04  
6.89693727e-04 5.62794914e-04 2.04202021e-04 5.72945646e-05  
1.38077536e-03 9.77320014e-04 6.86972111e-04 3.79844714e-04  
3.06843343e-04 2.39869347e-04 8.20080843e-03 1.64424375e-04  
2.43019123e-04 6.90443630e-05 1.02472673e-04 2.97914303e-05  
2.91632419e-03 1.00962468e-04 2.74828938e-03 1.74496003e-04  
9.85223451e-04 4.98584588e-04 1.13652623e-03 2.18112230e-01  
1.14814267e-01 1.30104506e-03 2.05928955e-04 3.65082902e-04  
3.04962276e-03 8.24127579e-04 1.23354164e-03 2.55797175e-03  
5.21821994e-03 5.38467255e-04 1.03299483e-03 1.22638242e-02  
2.99209845e-03 2.29965008e-04 8.97940323e-02 3.64774569e-05  
3.07429764e-05 7.33877998e-04 1.74680492e-04 4.56302514e-04  
2.22622906e-03 5.30592806e-04 3.67658125e-04 5.20422705e-04

Start coding or [generate](#) with AI.