

```
1 from keras.models import load_model # pip install
   tensorflow
2 from keras.layers import DepthwiseConv2D # pip
   install tensorflow
3 import cv2 # pip install opencv-python
4 import numpy as np # pip install numpy
5
6 # Disable scientific notation for clarity
7 np.set_printoptions(suppress=True)
8
9
10 class CustomDepthwiseConv2D(DepthwiseConv2D):
11     def __init__(self, **kwargs):
12         if 'groups' in kwargs:
13             del kwargs['groups']
14             super().__init__(**kwargs)
15
16 # Create a dictionary of custom objects to pass to
   the load_model function
17 custom_objects = {
18     'DepthwiseConv2D': CustomDepthwiseConv2D,
19 }
20
21
22 # Load the model
23 model = load_model("data/converted_keras/
   keras_model.h5", custom_objects=custom_objects,
   compile=False)
24
25 # Load the labels
26 class_names = open("data/converted_keras/labels.txt
   ", "r").readlines()
27
28 # CAMERA can be 0 or 1 based on default camera of
   your computer
29 camera = cv2.VideoCapture(1)
30
31 while True:
32     # Grab the webcam's image.
33     ret, image = camera.read()
34
35     # Resize the raw image into (224-height,224-
```

```
35 width) pixels
36     image = cv2.resize(image, (224, 224),
    interpolation=cv2.INTER_AREA)
37
38     # Show the image in a window
39     cv2.imshow("Webcam Image", image)
40
41     # Make the image a numpy array and reshape it
    to the models input shape.
42     image = np.asarray(image, dtype=np.float32).
    reshape(1, 224, 224, 3)
43
44     # Normalize the image array
45     image = (image / 127.5) - 1
46
47     # Predicts the model
48     prediction = model.predict(image)
49     index = np.argmax(prediction) # argmax()
    returns first occurrence of max value.
50     class_name = class_names[index]
51     confidence_score = prediction[0][index]
52
53     # Print prediction and confidence score
54     print("Class:", class_name[2:], end="")
55     print("Confidence Score:", str(np.round(
    confidence_score * 100))[:-2], "%")
56
57     # Listen to the keyboard for presses.
58     keyboard_input = cv2.waitKey(1)
59
60     # 27 is the ASCII for the esc key on your
    keyboard.
61     if keyboard_input == 27:
62         break
63
64 camera.release()
65 cv2.destroyAllWindows()
66
```