

```
1 #-----TensorFlow Imports-----#
2 import tensorflow.keras
3 from PIL import Image, ImageOps
4 import numpy as np
5
6 #-----Systemlink Imports-----#
7 import json, time
8 import requests
9
10 #-----Teachable Machines Model Function-----#
11 def Run_Model():
12     # Disable scientific notation for clarity
13     np.set_printoptions(suppress=True)
14
15     # Load the model
16     model = tensorflow.keras.models.load_model('keras_model.h5')
17
18     # Create the array of the right shape to feed into the keras model
19     data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
20
21     # Grab image saved on my desktop
22     image = Image.open('image.png')
23
24     #resize the image to a 224x224 with the same strategy as in TM2:
25     #resizing the image to be at least 224x224 and then cropping from the center
26     size = (224, 224)
27     image = ImageOps.fit(image, size, Image.ANTIALIAS)
28
29     #turn the image into a numpy array
30     image_array = np.asarray(image)
31
32     # display the resized image
33     image.show()
34
35     # Normalize the image
36     normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1
37
38     # Load the image into the array
39     data[0] = normalized_image_array
40
41     # run the inference
42     prediction = model.predict(data)
43     print(prediction)
44
45     #Finds index of highest Percent value
46     IndMax = np.argmax(prediction)
47     #IndMax = int(IndMax)
48     #print(type(IndMax))
49     print(IndMax)
50
51     if IndMax == int(0):
52         print('Found it!')
53         Put_SL('FoundTarget', 'BOOLEAN', 'true')
54
55     else:
56         print('Wrong Target')
57         Put_SL('FoundTarget', 'BOOLEAN', 'false')
```

```
58
59 #-----Tear Picture from Raspberry Pi-----#
60 url = 'Enter Pi URL here'
61
62 def Get_Image():
63     try:
64         data = requests.get(url)
65         with open('image.png', 'wb') as f:
66             f.write(data.content)
67     except Exception as e:
68         print(e)
69
70 #-----Systemlink Function Setup-----#
71 Key = 'Enter Key Here'
72
73 def SL_setup():
74     urlBase = "https://api.systemlinkcloud.com/nitag/v2/tags/"
75     headers = {"Accept": "application/json", "x-ni-api-key": Key}
76     return urlBase, headers
77
78 def Put_SL(Tag, Type, Value):
79     urlBase, headers = SL_setup()
80     urlValue = urlBase + Tag + "/values/current"
81     propValue = {"value": {"type": Type, "value": Value}}
82     try:
83         reply = requests.put(urlValue, headers=headers, json=propValue).text
84     except Exception as e:
85         print(e)
86         reply = 'failed'
87     return reply
88
89 def Get_SL(Tag):
90     urlBase, headers = SL_setup()
91     urlValue = urlBase + Tag + "/values/current"
92     try:
93         value = requests.get(urlValue, headers=headers).text
94         data = json.loads(value)
95         # print(data)
96         result = data.get("value").get("value")
97     except Exception as e:
98         print(e)
99         result = 'failed'
100     return result
101
102 def Create_SL(Tag, Type):
103     urlBase, headers = SL_setup()
104     urlTag = urlBase + Tag
105     propName = {"type": Type, "path": Tag}
106     try:
107         requests.put(urlTag, headers=headers, json=propName).text
108     except Exception as e:
109         print(e)
110
111
112
113
114
```

```
115 |
116 | #-----MAIN CODE-----#
117 | start = True
118 | while True:
119 |     #Waiting for reponse from EV3
120 |     if Get_SL('TakePicture') == 'true':
121 |
122 |         Put_SL('TakePicture', 'BOOLEAN', 'false') #resets the tag
123 |         print('Taking Picture')
124 |         Get_Image() #Function to take the picture from Pi
125 |
126 |         Run_Model() #Run image through the TensorFlow Model, determine if it's the target,
communicate to EV3
127 |
128 |     else:
129 |         # print('not yet')
130 |         continue #Loop waits until "TakePicture" tag is set to true
```