이미지 예측

```
val_datagen = ImageDataGenerator(rescale = 1.0/255.)

test_generator = val_datagen.flow_from_directory(
    'C:₩₩Users₩₩SAMSUNG₩₩Desktop₩₩MyPy₩₩project₩₩new_project₩₩test',
    target_size=(size, size),
    batch_size=batch_size,
    shuffle=False,
    class_mode='binary')

 # 초반부에 써놨지만 val이랑 test 이름만 바꿈
```

```
Found 721 images belonging to 6 classes.
```

```
model.evaluate(test_generator, steps=721 // batch_size)

# steps는 validation data 개수만큼 지정
```

```
45/45 [==============================] - 42s 927ms/step - loss: 0.1672 - accuracy: 0.9444
```

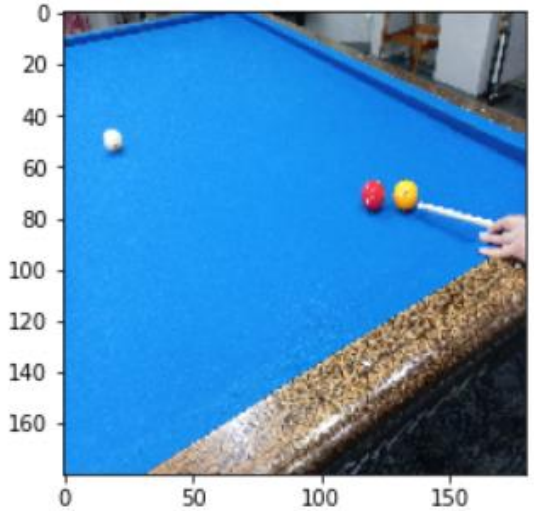
```
[0.16723312437534332, 0.9444444179534912]
```

steps는 validation data 개수만큼 지정

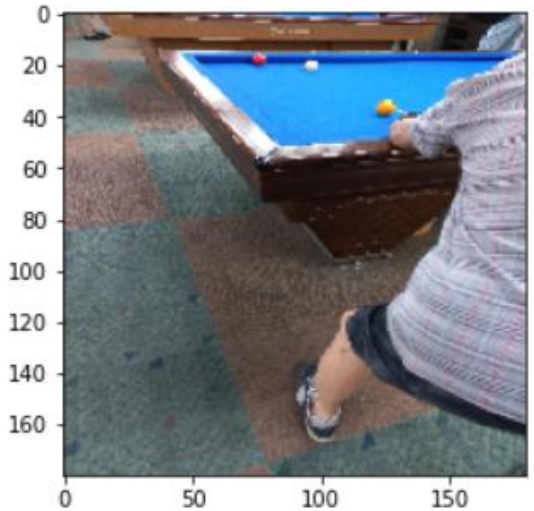
이미지 지정 후 예측 값 확인

```
np.set_printoptions(suppress=True)

from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt

img_path = 'predict_image_path'  # Billiard, Bowling, Cycling, Golf, MachineRunning,
Pingpong
img = image.load_img(img_path, target_size=(180, 180))
img_array = image.img_to_array(img)
plt.imshow(img_array/255)
expanded_img_array = np.expand_dims(img_array, axis=0)
preprocessed_img = expanded_img_array / 255

prediction = model2.predict(preprocessed_img)
print(np.array(prediction[0]))

# [Billiard  Bowling  Cycling  Golf  MachineRunning  Pingpong]
```

1. 당구



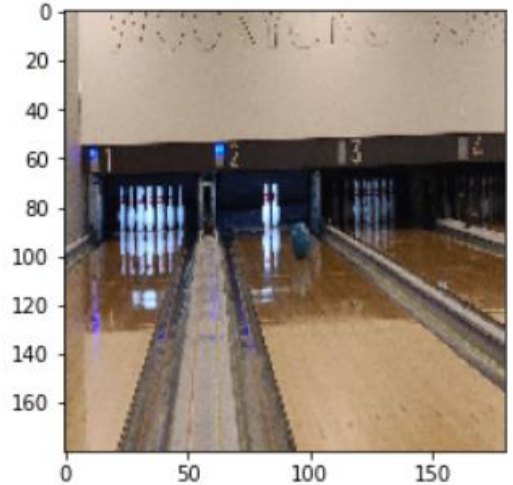
```
[1.         0.9995914  0.         0.         0.         0.42776412]
```



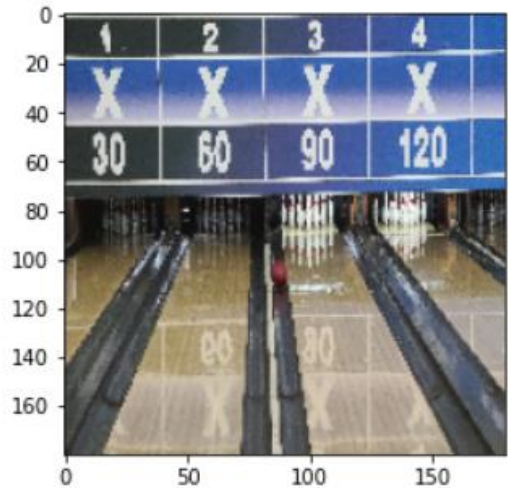
```
[0.9999998  0.05535468 0.         0.         0.         0.9868097 ]
```

서로 다르게 생긴 당구 이미지라도 모두 Billiard 클래스의 예측률이 가장 높음 (클래스 분류 성공)

2. 볼링



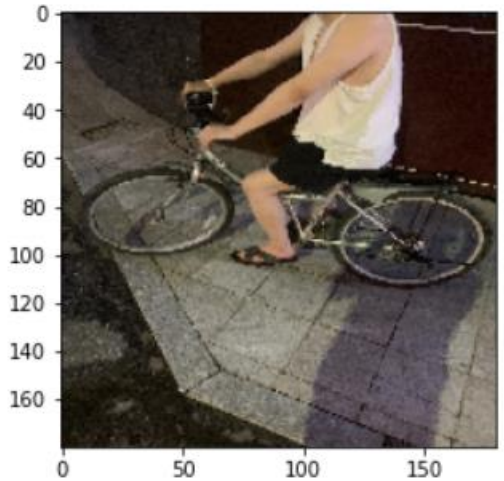
```
[0.00184122 0.99683034 0.00000037 0.35094672 0.00002263 0.15503743]
```



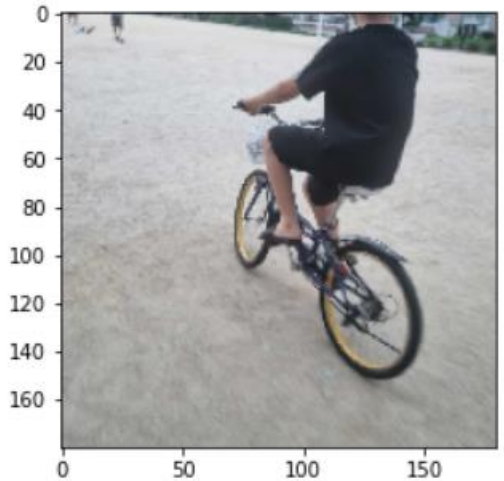

```
[0.00003574 0.99676836 0.00000359 0.00039539 0.00000005 0.0944784 ]
```

서로 다르게 생긴 볼링 이미지라도 모두 Bowling 클래스의 예측률이 가장 높음 (클래스 분류 성공)

3. 자전거



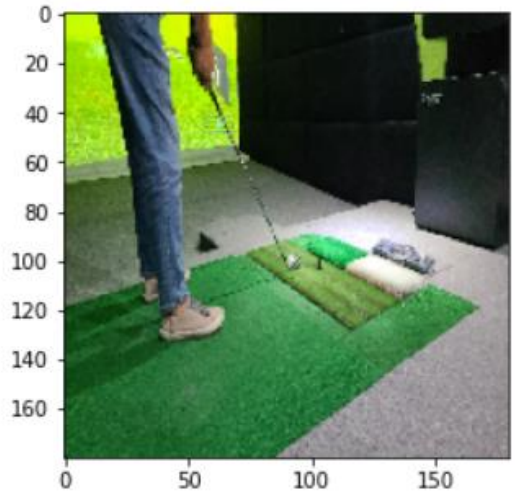
```
[0.00108001 0.3239639  0.9715294  0.7917694  0.22048551 0.00972053]
```



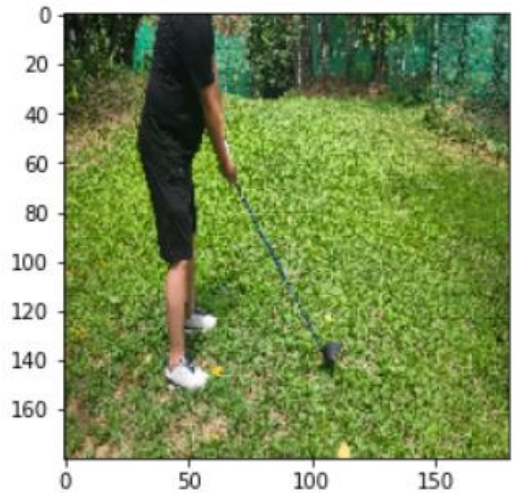
```
[0.00001991 0.6973912  0.97617126 0.84494    0.761382   0.00053573]
```

서로 다르게 생긴 자전거 이미지라도 모두 Cycling 클래스의 예측률이 가장 높음 (클래스 분류 성공)

4. 골프



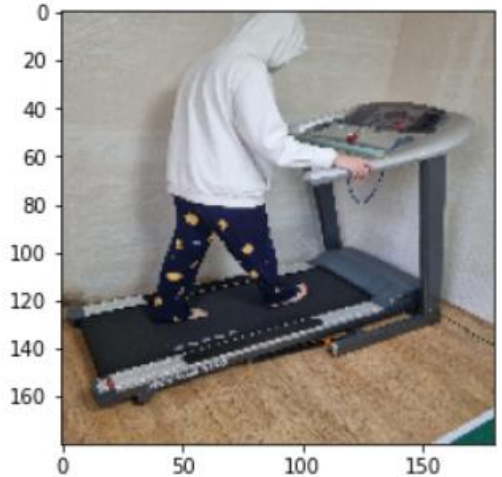

```
[0.00002109 0.8066366  0.00022909 0.9991114  0.00003228 0.26046598]
```



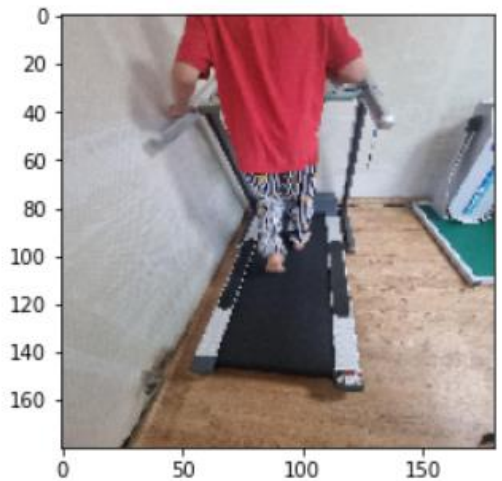

```
[0.000001   0.5777599  0.9973171  0.71749306 0.02596071 0.00006771]
```

두번째 골프 이미지에 경우 자전거 클래스로 분류 (클래스 분류 일부 성공)

4. 런닝머신



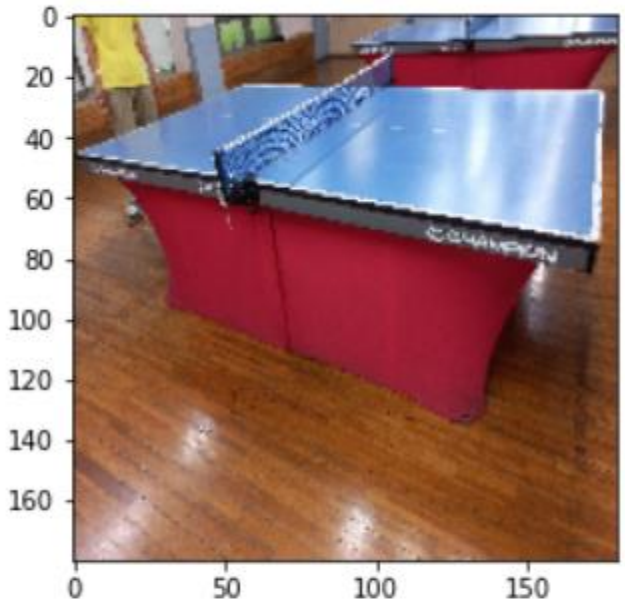
```
[0.00006472 0.82416606 0.0140352  0.5330569  0.9960704  0.00055158]
```



```
[0.00014174 0.6882539  0.01992747 0.0524604  0.999499   0.00011423]
```

서로 다르게 생긴 런닝머신 이미지라도 모두 MachineRunning 클래스의 예측률이 가장 높음 (클래스 분류 성공)

5. 탁구




```
[0.03032136 0.91186595 0.00000001 0.00228629 0.         0.9999475 ]
```

탁구는 이미지가 모두 비슷함 (클래스 분류 성공)