my-assign-3

November 17, 2023

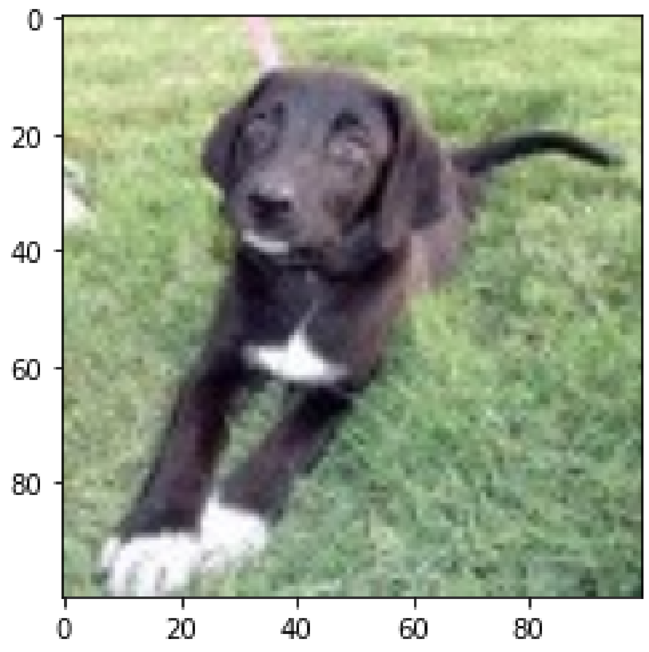
[25]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
import random
```

[26]:
```
x_train = np.loadtxt('C:/Users/HP/Desktop/DL_LAB/Public/PR 3 DL/Data/input.
  ↪csv',delimiter=',')
x_test = np.loadtxt('C:/Users/HP/Desktop/DL_LAB/Public/PR 3 DL/Data/input_test.
  ↪csv',delimiter=',')
y_train = np.loadtxt('C:/Users/HP/Desktop/DL_LAB/Public/PR 3 DL/Data/labels.
  ↪csv',delimiter=',')
y_test = np.loadtxt('C:/Users/HP/Desktop/DL_LAB/Public/PR 3 DL/Data/labels_test.
  ↪csv',delimiter=',')
```

[27]:
```
x_train = x_train.reshape(len(x_train),100,100,3)
y_train = y_train.reshape(len(y_train),1)
x_test = x_test.reshape(len(x_test),100,100,3)
y_test = y_test.reshape(len(y_test),1)
```

[28]:
```
x_train, x_test = x_train/255, x_test/255
```

[29]:
```
idx = random.randint(0,len(x_train))
plt.imshow(x_train[idx])
plt.show()
```



[30]:
```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Flatten,Dense
```

[31]:
```python
model = Sequential([
    Conv2D(32,(3,3),activation='relu',input_shape=(100,100,3)),
    MaxPooling2D((2,2)),

    Conv2D(32,(3,3),activation='relu'),
    MaxPooling2D((2,2)),

    Flatten(),
    Dense(64,activation='relu'),
    Dense(1,activation='sigmoid')
])
```

[54]:
```python
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

[55]:
```python
model.fit(x_train,y_train,epochs=10)
```

```
Epoch 1/10
63/63 [==============================] - 17s 240ms/step - loss: 0.0731 -
accuracy: 0.9765
Epoch 2/10
63/63 [==============================] - 15s 236ms/step - loss: 0.0564 -
accuracy: 0.9815
Epoch 3/10
```

```
63/63 [==============================] - 14s 229ms/step - loss: 0.0267 -
accuracy: 0.9950
Epoch 4/10
63/63 [==============================] - 15s 240ms/step - loss: 0.0201 -
accuracy: 0.9955
Epoch 5/10
63/63 [==============================] - 15s 241ms/step - loss: 0.0523 -
accuracy: 0.9835
Epoch 6/10
63/63 [==============================] - 16s 248ms/step - loss: 0.0467 -
accuracy: 0.9835
Epoch 7/10
63/63 [==============================] - 16s 252ms/step - loss: 0.0307 -
accuracy: 0.9925
Epoch 8/10
63/63 [==============================] - 16s 248ms/step - loss: 0.0110 -
accuracy: 0.9980
Epoch 9/10
63/63 [==============================] - 15s 240ms/step - loss: 0.0057 -
accuracy: 1.0000
Epoch 10/10
63/63 [==============================] - 15s 241ms/step - loss: 0.0034 -
accuracy: 1.0000
```

```
[55]: <keras.src.callbacks.History at 0x1893ec09280>
```

```
[56]: model.evaluate(x_test,y_test)
```

```
13/13 [==============================] - 1s 70ms/step - loss: 2.2897 - accuracy:
0.6775
```

```
[56]: [2.289689064025879, 0.6775000095367432]
```

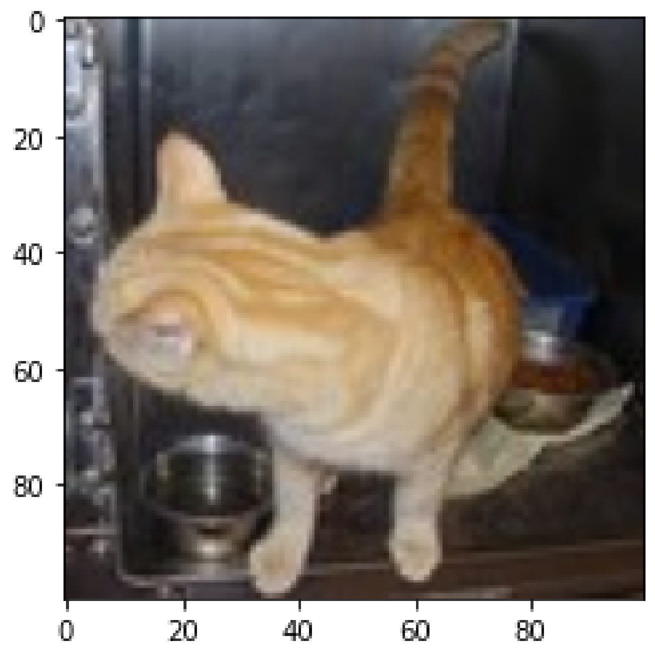
```
[53]: idx2 = random.randint(0,len(y_test))
plt.imshow(x_test[idx2,:])
plt.show()

y_pred = model.predict(x_test[idx2,:].reshape(1,100,100,3))
y_pred = y_pred > 0.5

if(y_pred == 0):
    pred = 'Dog'

else:
    pred = 'Cat'

print("Our Model Says it is: ",pred)
```



```
1/1 [==============================] - 0s 60ms/step
Our Model Says it is:  Cat
```

[ ]:

[ ]: