

cat.1744.jpg ×

...



```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
IMG_SIZE=224
BATCH_SIZE=32
```

```
train_datagen = ImageDataGenerator(rescale=1./255,validation_split=0.2)
```

```
train_generator = train_datagen.flow_from_directory( '/content/drive/MyDrive,
target_size=(IMG_SIZE, IMG_SIZE),
batch_size=BATCH_SIZE,
class_mode='binary',
subset='training')
```

```
Found 6404 images belonging to 2 classes.
```

```
print(train_generator.class_indices)
```

```
{'cats': 0, 'dogs': 1}
```

```
val_generator = train_datagen.flow_from_directory( '/content/drive/MyDrive/II
target_size=(IMG_SIZE, IMG_SIZE),
batch_size=BATCH_SIZE,
class_mode='binary',
subset='validation')
```

```
Found 1601 images belonging to 2 classes.
```

```
model=keras.Sequential([
    layers.Conv2D(32,(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(128,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid')
])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/b
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```



```
model.summary()
```

What can I help you build?



Model: "sequential"

Layer (type)	Output Shape	
conv2d (Conv2D)	(None, 222, 222, 32)	
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	
conv2d_1 (Conv2D)	(None, 109, 109, 64)	
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	
conv2d_2 (Conv2D)	(None, 52, 52, 128)	
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	
flatten (Flatten)	(None, 86528)	
dense (Dense)	(None, 128)	11
dense_1 (Dense)	(None, 1)	

Total params: 11,169,089 (42.61 MB)

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
model.fit(train_generator,epochs=5,validation_data=val_generator,batch_size=128)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters,
self._warn_if_super_not_called()
Epoch 1/5
201/201 ━━━━━━━━━━━ 0s 5s/step - accuracy: 0.5426 - loss: 0.795
self._warn_if_super_not_called()
201/201 ━━━━━━━━━━━ 1141s 5s/step - accuracy: 0.5428 - loss: 0.
Epoch 2/5
201/201 ━━━━━━━━━━━ 1232s 6s/step - accuracy: 0.6823 - loss: 0.
Epoch 3/5
201/201 ━━━━━━━━━━━ 796s 4s/step - accuracy: 0.7403 - loss: 0.5
Epoch 4/5
201/201 ━━━━━━━━━━━ 807s 4s/step - accuracy: 0.7768 - loss: 0.4
Epoch 5/5
201/201 ━━━━━━━━━━━ 810s 4s/step - accuracy: 0.8480 - loss: 0.3
<keras.src.callbacks.history.History at 0x78fcee76d50>
```

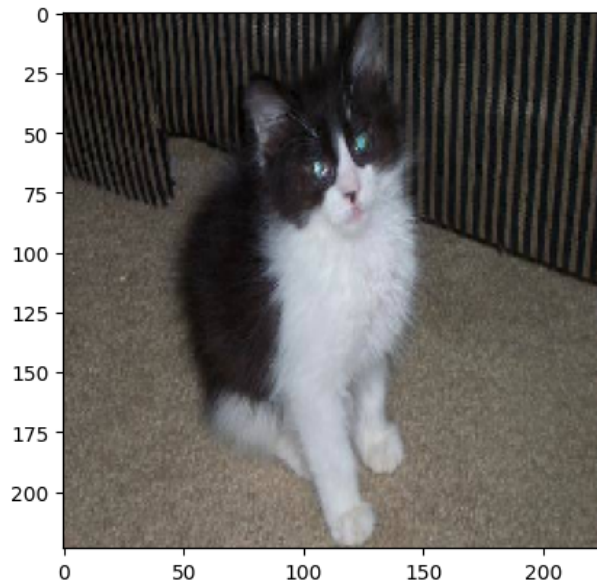
```
model.save('/content/drive/MyDrive/INTERNSHIP/training_set/DOG_VS_CAT.h5')
```

WARNING:absl:You are saving your model as an HDF5 file via `model.save()`

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
model=load_model("/content/drive/MyDrive/INTERNSHIP/training_set/DOG_VS_CAT.h5")
print('Model Loaded')
```

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be computed.

```
test_image_path="/content/drive/MyDrive/INTERNSHIP/CATVSDOG/CAT/cat.129.jpg"
img=image.load_img(test_image_path,target_size=(224,224))
plt.imshow(img)
plt.axis()
plt.show()
```



```
img_array=image.img_to_array(img)
img_array=np.expand_dims(img_array,axis=0)
img_array /= 255.
```

```
prediction = model.predict(img_array)
```



1/1 — 0s 69ms/step

```
print(prediction)
if prediction <= 0.5:
    print("The Given Image Is CAT")
else:
    print("The Given Image Is DOG")
```



[[0.08043668]]
The Given Image Is CAT