

```

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

Found 93 images belonging to 6 classes.
Found 20 images belonging to 6 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')
])

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

model.fit(train_generator, epochs=5, validation_data=val_generator)

Epoch 1/5
3/3 [=====] - 33s 13s/step - loss: 2.6246 -
accuracy: 0.7312 - val_loss: 0.9264 - val_accuracy: 0.1500
Epoch 2/5
3/3 [=====] - 14s 4s/step - loss: 0.7397 -
accuracy: 0.1828 - val_loss: 0.8253 - val_accuracy: 0.1500

```

```
Epoch 3/5
3/3 [=====] - 13s 4s/step - loss: 0.5946 -
accuracy: 0.3333 - val_loss: 0.7525 - val_accuracy: 0.6500
Epoch 4/5
3/3 [=====] - 13s 4s/step - loss: 0.2070 -
accuracy: 0.7419 - val_loss: 1.4361 - val_accuracy: 0.4500
Epoch 5/5
3/3 [=====] - 13s 4s/step - loss: -0.6286 -
accuracy: 0.6667 - val_loss: 2.3370 - val_accuracy: 0.6000
```

```
<keras.src.callbacks.History at 0x7a26abc5a620>
```

```
model.save("chessman_modal.h5", "label.text")
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/
training.py:3103: UserWarning: You are saving your model as an HDF5
file via `model.save()`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')`.
  saving_api.save_model(
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
```

```
model = load_model('/content/chessman_modal.h5')
```

```
test_image =
image.load_img('/content/drive/MyDrive/chessman_image_dataset/King/
00000000.jpg', target_size=(224, 224))
```

```
img = image.img_to_array(test_image)
img = np.expand_dims(img, axis=0)
```

```
img /= 255.
prediction = model.predict(img)
print(prediction)
```

```
1/1 [=====] - 0s 112ms/step
[[0.26071528]]
```

```
if prediction < 0.5:
    print("Prediction: This is king(Probability:", prediction[0][0])
else:
    print("Prediction: This is rook(Probability:", prediction[0][0])
```

```
Prediction: This is king(Probability: 0.26071528
```