

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

from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
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/COVID-19 chest X-ray dataset',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)
```

Found 76 images belonging to 2 classes.

```
val_generator=train_datagen.flow_from_directory(
    '/content/drive/MyDrive/COVID-19 chest X-ray dataset',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

Found 18 images belonging to 2 classes.

```
class_names=list(train_generator.class_indices.keys())
print(class_names)
print(train_generator.class_indices)
```

```
['covid', 'normal']
{'covid': 0, 'normal': 1}
```

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

```
model = tf.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(len(class_names), activation='softmax')
])

```

/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

```

super().__init__(activity_regularizer=activity_regularizer,
**kwargs)

```

```
model.summary()
```

Model: "sequential"

Layer (type) Param #	Output Shape	
conv2d (Conv2D) 896	(None, 222, 222, 32)	
max_pooling2d (MaxPooling2D) 0	(None, 111, 111, 32)	
conv2d_1 (Conv2D) 18,496	(None, 109, 109, 64)	
max_pooling2d_1 (MaxPooling2D) 0	(None, 54, 54, 64)	
conv2d_2 (Conv2D) 73,856	(None, 52, 52, 128)	
max_pooling2d_2 (MaxPooling2D) 0	(None, 26, 26, 128)	
flatten (Flatten)	(None, 86528)	

0				
	dense (Dense)		(None, 128)	
11,075,712				
	dense_1 (Dense)		(None, 4)	
516				

Total params: 11,169,476 (42.61 MB)

Trainable params: 11,169,476 (42.61 MB)

Non-trainable params: 0 (0.00 B)

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

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

Epoch 1/5

3/3 ————— 0s 2s/step - accuracy: 0.5090 - loss: 1.3789

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/
data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset`
class should call `super().__init__(**kwargs)` in its constructor.
`**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will
be ignored.
```

```
self._warn_if_super_not_called()
```

3/3 ————— 28s 8s/step - accuracy: 0.5331 - loss: 1.4404 - val_accuracy: 0.2778 - val_loss: 1.8378

Epoch 2/5

3/3 ————— 13s 3s/step - accuracy: 0.2658 - loss: 1.4800 - val_accuracy: 0.7222 - val_loss: 0.6103

Epoch 3/5

3/3 ————— 10s 4s/step - accuracy: 0.7586 - loss: 0.5771 - val_accuracy: 0.7222 - val_loss: 0.5925

Epoch 4/5

3/3 ————— 11s 3s/step - accuracy: 0.7239 - loss: 0.5789 - val_accuracy: 0.7222 - val_loss: 0.5418

Epoch 5/5

3/3 ————— 12s 5s/step - accuracy: 0.7586 - loss: 0.4919 - val_accuracy: 0.7222 - val_loss: 0.4803

<keras.src.callbacks.history.History at 0x7d60eca4c990>

```
model.save('/content/drive/MyDrive/COVID-19 chest X-ray dataset.h5')
```

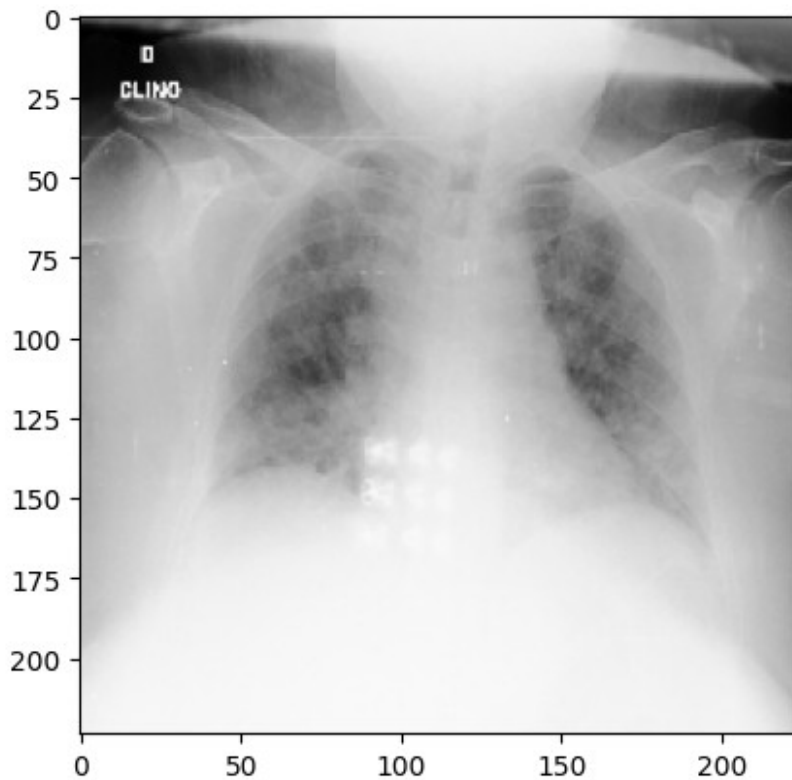
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

```
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/COVID-19 chest X-ray
dataset.h5')
print("Model Loaded Successfully")
```

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.

Model Loaded Successfully

```
test_image_path="/content/drive/MyDrive/COVID-19 chest X-ray
dataset/covid/01E392EE-69F9-4E33-BFCE-E5C968654078.jpeg"
img=image.load_img(test_image_path,target_size=(224,224,3))
plt.imshow(img)
plt.axis()
plt.show()
```



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

```
prediction = model.predict(img_array)
ind=np.argmax(prediction)
prediction=prediction[0][ind]
```

```
1/1 _____ 0s 173ms/step
```

```
prediction=model.predict(img_array)
print(prediction)
ind=np.argmax(prediction)
print(class_names[ind])
```

```
1/1 _____ 0s 68ms/step
[[0.8219508  0.17804915]]
covid
```