



```
In [1]: pip install opendatasets
```

```
Collecting opendatasets
  Downloading opendatasets-0.1.22-py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from opendatasets) (4.67.1)
Requirement already satisfied: kaggle in /usr/local/lib/python3.11/dist-packages (from opendatasets) (1.7.4.5)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from opendatasets) (8.2.1)
Requirement already satisfied: bleach in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (6.2.0)
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2025.4.26)
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (3.4.2)
Requirement already satisfied: idna in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (3.10)
Requirement already satisfied: protobuf in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (5.29.5)
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2.9.0.post0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (8.0.4)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2.32.3)
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (75.2.0)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (1.17.0)
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (1.3)
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (2.4.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (from kaggle->opendatasets) (0.5.1)
Downloading opendatasets-0.1.22-py3-none-any.whl (15 kB)
Installing collected packages: opendatasets
Successfully installed opendatasets-0.1.22
```

```
In [2]: import tensorflow as tf
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.models import load_model
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Flatten,Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import pandas as pd
import opendatasets as od
```

```
In [3]: od.download("https://www.kaggle.com/datasets/jtiptj/chest-xray-pneumoniacovid")
```

Please provide your Kaggle credentials to download this dataset. Learn more: <http://bit.ly/kaggle-creds>  
Your Kaggle username: mohanapriya9080  
Your Kaggle Key: .....  
Dataset URL: <https://www.kaggle.com/datasets/jtiptj/chest-xray-pneumoniacovid19-tuberculosis>  
Downloading chest-xray-pneumoniacovid19tuberculosis.zip to ./chest-xray-pneumoniacovid19tuberculosis

100%|██████████| 1.74G/1.74G [00:17<00:00, 108MB/s]

```
In [4]: data_dir = "/content/test-dataset"
```

```
In [5]: IMG_SIZE=224  
        BATCH_SIZE=32
```

```
In [6]: train_datagen=ImageDataGenerator(rescale=1./255,validation_split=0.2)
```

```
In [7]: train_generator=train_datagen.flow_from_directory('/content/chest-xray-pneumoniacovid19tuberculosis',  
                                                         target_size=(IMG_SIZE, IMG_SIZE),  
                                                         batch_size=BATCH_SIZE,  
                                                         class_mode='categorical')  
Found 5061 images belonging to 4 classes.
```

```
In [8]: val_generator=train_datagen.flow_from_directory('/content/chest-xray-pneumonia',  
                                                       target_size=(IMG_SIZE, IMG_SIZE),  
                                                       batch_size=BATCH_SIZE,  
                                                       class_mode='categorical')  
Found 1265 images belonging to 4 classes.
```

```
In [9]: class_indices = train_generator.class_indices  
        class_names = list(class_indices.keys())  
        print("Class Indices:", class_indices)  
        print("Class Names:", class_names)  
        print(class_indices)
```

Class Indices: {'COVID19': 0, 'NORMAL': 1, 'PNEUMONIA': 2, 'TURBERCULOSIS': 3}  
Class Names: ['COVID19', 'NORMAL', 'PNEUMONIA', 'TURBERCULOSIS']  
{'COVID19': 0, 'NORMAL': 1, 'PNEUMONIA': 2, 'TURBERCULOSIS': 3}

```
In [10]: 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(4, 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)

```
In [11]: model.summary()
```

Model: "sequential"

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

```
In [12]: model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
model.fit(train_generator,epochs=4,validation_data=val_generator,batch_size=BA
```

```
/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()
```

Epoch 1/4

159/159 ————— 601s 4s/step - accuracy: 0.7475 - loss: 0.8039 - val\_accuracy: 0.8909 - val\_loss: 0.3004

Epoch 2/4

159/159 ————— 577s 4s/step - accuracy: 0.9381 - loss: 0.1793 - val\_accuracy: 0.9368 - val\_loss: 0.1731

Epoch 3/4

159/159 ————— 622s 4s/step - accuracy: 0.9592 - loss: 0.1208 - val\_accuracy: 0.9296 - val\_loss: 0.2143

Epoch 4/4

159/159 ————— 621s 4s/step - accuracy: 0.9735 - loss: 0.0855 - val\_accuracy: 0.9431 - val\_loss: 0.1674

```
Out[12]: <keras.src.callbacks.history.History at 0x7ab3a0f6e210>
```

```
In [13]: model.save('/content/chest-xray-pneumoniacovid19tuberculosis/train/xray.h5')
```

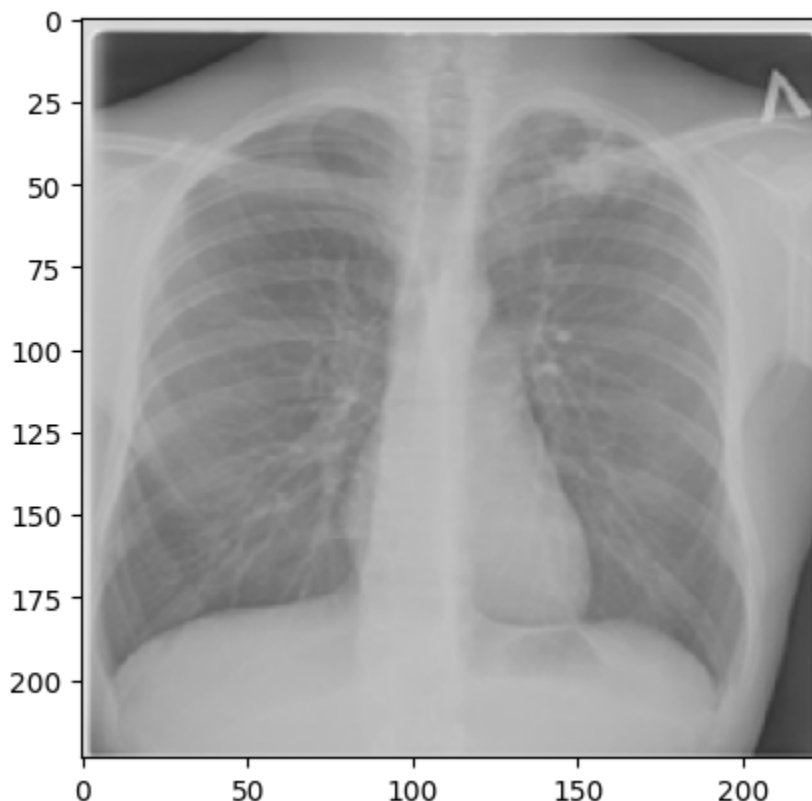
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save\_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')` or `keras.saving.save\_model(model, 'my\_model.keras')`.

```
In [14]: 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/chest-xray-pneumoniacovid19tuberculosis/train/xray')
         print("Model Loaded")
```

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

```
In [15]: test_image_path="/content/chest-xray-pneumoniacovid19tuberculosis/train/TURBER
         img=image.load_img(test_image_path,target_size=(224,224))
         plt.imshow(img)
         plt.axis()
         plt.show()
```



```
In [16]: img_array=image.img_to_array(img)
         img_array=np.expand_dims(img_array,axis=0)
         img_array=img_array/255.0
```

```
In [17]: prediction=model.predict(img_array)
```

```
ind=np.argmax(prediction[0])  
print(class_names[ind])
```

**1/1**  **0s** 142ms/step  
TURBERCULOSIS