


```
from google.colab import files
uploaded = files.upload()
```

 Choose Files animal\_dataset.zip


- **animal\_dataset.zip**(application/zip) - 38014450 bytes, last modified: 7/23/2025 - 100% done

Saving animal\_dataset.zip to animal\_dataset.zip

```
import zipfile
import os
```

```
# Unzip the file
with zipfile.ZipFile("animal_dataset.zip", 'r') as zip_ref:
    zip_ref.extractall("animal_dataset")
```

```
# Check folders
print("✅ Extracted folders:", os.listdir("animal_dataset"))
```

 ✅ Extracted folders: ['dataset']

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


```
# Define paths
data_dir = "/content/animal_dataset/dataset"
```

```
# Image parameters
img_height = 224
img_width = 224
batch_size = 32
```

```
# Split data into training and validation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
)
```

```
train_generator = train_datagen.flow_from_directory(
    data_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    subset='training'
)
```

```
val_generator = train_datagen.flow_from_directory(
    data_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation'
)
```

 Found 1561 images belonging to 15 classes.  
Found 383 images belonging to 15 classes.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
```

```
# Build the CNN
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(224, 224, 3)),
    MaxPooling2D(2,2),

    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Conv2D(128, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dropout(0.5),
    Dense(1000, activation='relu')
])
```

◆ What can I help you build?



```

Dense(128, activation= relu ),
Dense(15, activation='softmax') # 15 classes
])

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

# Train the model
history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=10
)

```

```

/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape` to `input_shape` in the constructor of `Conv2D` or `Conv3D` layers. It is deprecated and will be removed in a future version.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/10
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class does not implement the `get_data_adapter` method. This may lead to unexpected behavior.
self._warn_if_super_not_called()
49/49 ━━━━━━━━━━━ 191s 4s/step - accuracy: 0.0855 - loss: 2.9473 - val_accuracy: 0.2167 - val_loss: 2.5269
Epoch 2/10
49/49 ━━━━━━━━━━━ 187s 4s/step - accuracy: 0.2735 - loss: 2.2597 - val_accuracy: 0.2768 - val_loss: 2.2861
Epoch 3/10
49/49 ━━━━━━━━━━━ 188s 4s/step - accuracy: 0.5480 - loss: 1.5089 - val_accuracy: 0.3316 - val_loss: 2.5306
Epoch 4/10
49/49 ━━━━━━━━━━━ 188s 4s/step - accuracy: 0.7772 - loss: 0.7780 - val_accuracy: 0.3081 - val_loss: 3.0255
Epoch 5/10
49/49 ━━━━━━━━━━━ 186s 4s/step - accuracy: 0.8951 - loss: 0.4045 - val_accuracy: 0.3499 - val_loss: 3.9105
Epoch 6/10
49/49 ━━━━━━━━━━━ 202s 4s/step - accuracy: 0.9453 - loss: 0.2454 - val_accuracy: 0.3264 - val_loss: 4.8202
Epoch 7/10
49/49 ━━━━━━━━━━━ 187s 4s/step - accuracy: 0.9766 - loss: 0.1358 - val_accuracy: 0.3446 - val_loss: 4.6841
Epoch 8/10
49/49 ━━━━━━━━━━━ 202s 4s/step - accuracy: 0.9915 - loss: 0.0707 - val_accuracy: 0.3368 - val_loss: 5.8086
Epoch 9/10
49/49 ━━━━━━━━━━━ 186s 4s/step - accuracy: 0.9836 - loss: 0.1093 - val_accuracy: 0.3394 - val_loss: 6.3553
Epoch 10/10
49/49 ━━━━━━━━━━━ 187s 4s/step - accuracy: 0.9817 - loss: 0.0993 - val_accuracy: 0.2924 - val_loss: 6.0163

```