**Classification:**

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

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

from tensorflow.keras.optimizers import Adam

# Set parameters

img\_height, img\_width = 128, 128

batch\_size = 32

num\_classes = len(os.listdir("animal\_data/train"))

# Data Generators with Augmentation for training

train\_datagen = ImageDataGenerator(

rescale=1./255,

rotation\_range=30,

zoom\_range=0.2,

horizontal\_flip=True

)

val\_datagen = ImageDataGenerator(rescale=1./255)

train\_generator = train\_datagen.flow\_from\_directory(

"animal\_data/train",

target\_size=(img\_height, img\_width),

batch\_size=batch\_size,

class\_mode='categorical'

)

val\_generator = val\_datagen.flow\_from\_directory(

"animal\_data/val",

target\_size=(img\_height, img\_width),

batch\_size=batch\_size,

class\_mode='categorical'

)

# Build CNN model

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(img\_height, img\_width, 3)),

MaxPooling2D(2, 2),

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

MaxPooling2D(2, 2),

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

MaxPooling2D(2, 2),

Flatten(),

Dense(256, activation='relu'),

Dropout(0.5),

Dense(num\_classes, activation='softmax')

])

# Compile the model

model.compile(optimizer=Adam(), loss='categorical\_crossentropy', metrics=['accuracy'])

# Train the model

history = model.fit(

train\_generator,

epochs=5,

validation\_data=val\_generator

)

# Save the model

model.save("animal\_classifier\_model.h5")