DOGS VS CATS CLASSIFICATION

Steps:

1. **Data Preparation**
2. **Model Building**
3. **Model Training**
4. **Model Evaluation**
5. **Model Deployment**

CODE:

```
import os

import numpy as np

import tensorflow as tf

from tensorflow.keras.preprocessing.image import
ImageDataGenerator

from tensorflow.keras import layers, models

from tensorflow.keras.preprocessing import image

from flask import Flask, request, jsonify
```

```
# Step 1: Data Preparation
train_dir = 'data/train'
validation_dir = 'data/validation'
train_datagen = ImageDataGenerator(rescale=1./255)
validation_datagen =
ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150, 150),
    batch_size=20,
    class_mode='binary'
)
validation_generator =
validation_datagen.flow_from_directory(
    validation_dir,
    target_size=(150, 150),
    batch_size=20,
    class_mode='binary'
)
```

```
# Step 2: Model Building
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(150, 150, 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(512, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])

model.compile(loss='binary_crossentropy',
optimizer='adam', metrics=['accuracy'])
# Step 3: Model Training
model.fit(
    train_generator,
    steps_per_epoch=100,
    epochs=15,
    validation_data=validation_generator,
    validation_steps=50
)
# Save the model
model.save('cats_and_dogs_classifier.h5')
# Step 4: Deployment using Flask
app = Flask(__name__)
model = tf.keras.models.load_model('cats_and_dogs_classifier.h5')
def prepare_image(file):
    img = image.load_img(file, target_size=(150, 150))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.0
    return img_array
@app.route('/predict', methods=['POST'])
```

```
def predict():
    if 'file' not in request.files:
        return jsonify({'error': 'No file provided'}), 400
    file = request.files['file']
    img_array = prepare_image(file)
    prediction = model.predict(img_array)
    result = 'dog' if prediction[0][0] > 0.5 else 'cat'
    return jsonify({'prediction': result})
if __name__ == '__main__':
    app.run(debug=True)
```

Instructions for Running the Script

1. **Prepare Dataset:**
   - Download the "Dogs vs. Cats" dataset.
   - Organize the dataset into data/train and data/validation directories.
2. **Save the Script:**
   - Save the entire script in a file named cats_and_dogs_classifier.py.

3. **Install Required Packages:**
    - Ensure you have the necessary Python packages installed:

```
pip install tensorflow flask
```

4. **Run the Script:**
    - Run the script using Python:

```
python cats_and_dogs_classifier.py
```

5. **Send a POST Request:**
    - Use a tool like Postman or cURL to send a POST request to http://127.0.0.1:5000/predict with an image file.