TASK 6

SUBJECT:

Programming For AI

PROGRAM:

BS DATA SCIENCE

SUBMITTED TO:

Sir Rasikh Ali

SUBMITTED BY:

FIZZA FAROOQ

ROLL NUMBER:

SU92-BSDSM-F23-017

BSDS (4A)

Task 6

Animal Herd Detection

Explanation

1. Importing Necessary Libraries

```
import cv2
import numpy as np
from geopy.distance import geodesic
import matplotlib.pyplot as plt
import folium
import re
```

- `cv2`: OpenCV is used for image processing and deep learning-based object detection.
- `numpy`: A numerical computing library used for mathematical operations and array manipulations.
- `geopy.distance`: Helps calculate the geographical distance between two locations.
- `matplotlib.pyplot`: Used for visualizing images with bounding boxes.
- `folium`: Used for generating interactive maps to display herd locations.
- `re`: Regular expressions module for extracting coordinates from Google Maps links.

2. Loading the YOLOv3 Model

```
net = cv2.dnn.readNet(r"C:\Users\M.A Computer\Desktop\animal herd detection\lab6\yolov3.weights",
                      r"C:\Users\M.A Computer\Desktop\animal herd detection\lab6\yolov3.cfg")
```

`cv2.dnn.readNet()`: Loads the YOLOv3 pre-trained weights and configuration file for object detection.

```
layersnames = net.getLayerNames()
output_layers = [layersnames[i - 1] for i in net.getUnconnectedOutLayers()]
with open("animal.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]
```

- Retrieves the names of all layers in the YOLOv3 network.
- `getUnconnectedOutLayers()`: Identifies the output layers of the model that perform detection.

- Reads `animal.names`, a file containing names of detected animal classes (e.g., "dog").
- `strip()` removes newline characters.

3. Function to Detect Animals

```
def animals_detect(frame):
    height, width, channels = frame.shape
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(output_layers)
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5 and classes[class_id] == "dog":
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
                w = int(detection[2] * width)
                h = int(detection[3] * height)
                x = int(center_x - w / 2)
                y = int(center_y - h / 2)
```

```
                boxes.append([x, y, w, h])
                confidences.append(float(confidence))
                class_ids.append(class_id)
    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
    return boxes, indexes
```

- A function that processes an image to detect animals.
- Extracts image dimensions (height, width, and color channels).
- Converts the image into a format suitable for YOLOv3 using `blobFromImage()`.
- Resizes the image to `416x416`, normalizes pixel values, and converts it into a format suitable for deep learning.
- Sets the processed image as input for YOLOv3.
- Performs forward propagation to obtain detection results.
- Initializes lists to store detected object class IDs, confidence scores, and bounding boxes.
- Iterates through detected objects.
- Extracts confidence scores and determines the class with the highest confidence.
- Filters out low-confidence detections and only processes "dog" class.
- Converts YOLOv3 normalized coordinates to actual pixel values.
- Calculates the top-left corner of the bounding box.
- Stores bounding box details, confidence, and class ID.
- Performs Non-Maximum Suppression (NMS) to remove duplicate overlapping detections.

3. Distance Calculation & Alerts

```
def calculate_the_distance(loc1, loc2):
    return geodesic(loc1, loc2).meters
def send_alert(distance):
    print(f"Alert! Animal herd detected nearbysuccesfully: Distance: {distance:.2f} meters")
```

- Calculates the real-world distance (in meters) between two geographical coordinates using `geopy.distance.geodesic()`.
- Prints an alert message when animals are detected near a specified location.

4. Extracting Coordinates from Google Maps Link

```
def extract_coordinates_from_place_link(link):
    match = re.search(r"@(-?\d+\.\d+),(-?\d+\.\d+)", link)
    if match:
        return float(match.group(1)), float(match.group(2))
    else:
        raise ValueError(" Google Maps place link is invalid. Please provide a valid link.")
```

- Extracts latitude and longitude from a Google Maps link using a regular expression.
- Returns the extracted coordinates as floating-point numbers.

6. Getting User Location Input

```
herd_locations = []

location_input = input("Enter your location (Google Maps place link or latitude,longitude):https://maps.app.goo.

if "google.com/maps/place" in location_input:
    current_location = extract_coordinates_from_place_link(location_input)
else:
    try:
        lat, lon = list(__builtins__.map(float, location_input.split(',')))
        current_location = (lat, lon)
    except ValueError:
        print("Invalid input. Please provide a valid Google Maps place link or latitude,longitude.")
        exit()
```

- Initializes an empty list to store detected herd locations.
- Prompts the user to input a location via a Google Maps link or latitude/longitude values.
- Extracts coordinates from a Google Maps link if detected.
- Parses latitude and longitude if entered directly.
- Exits if input is invalid.

7. Processing the Image for Detection

```
image_path = r"dog.jpg"
frame = cv2.imread(image_path)

if frame is None:
    print("Error: Could not load image")
else:
    boxes, indexes = animals_detect(frame)
    for i in range(len(boxes)):
        if i in indexes:
            herd_lat = current_location[0] + np.random.uniform(-0.01, 0.01)
            herd_lon = current_location[1] + np.random.uniform(-0.01, 0.01)
            herd_locations.append((herd_lat, herd_lon))
            distance = calculate_the_distance(current_location, (herd_lat, herd_lon))
            send_alert(distance)
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

- Loads an image (e.g., "dog.jpg") and checks for successful loading.
- Calls the animal detection function to obtain bounding boxes and detection indexes.
- Generates random herd locations near the current location for each detected object.
- Calculates the distance from the current location to each herd location and triggers alerts accordingly.

8. Displaying Results

```
    plt.figure(figsize=(10, 10))
    plt.imshow(frame_rgb)
    plt.axis('off')
    plt.title("Detected Animals")
    plt.show()
    folium_map = folium.Map(location=current_location, zoom_start=15)
    folium.Marker(
        location=current_location,
        popup="Your Location",
        icon=folium.Icon(color="blue")
    ).add_to(folium_map)
    for herd_loc in herd_locations:
        folium.Circle(
            location=herd_loc,
            radius=50,
            color="red",
            fill=True,
            fill_color="red",
            popup=f"Herd Location: {herd_loc}"
        ).add_to(folium_map)
    display(folium_map)
```

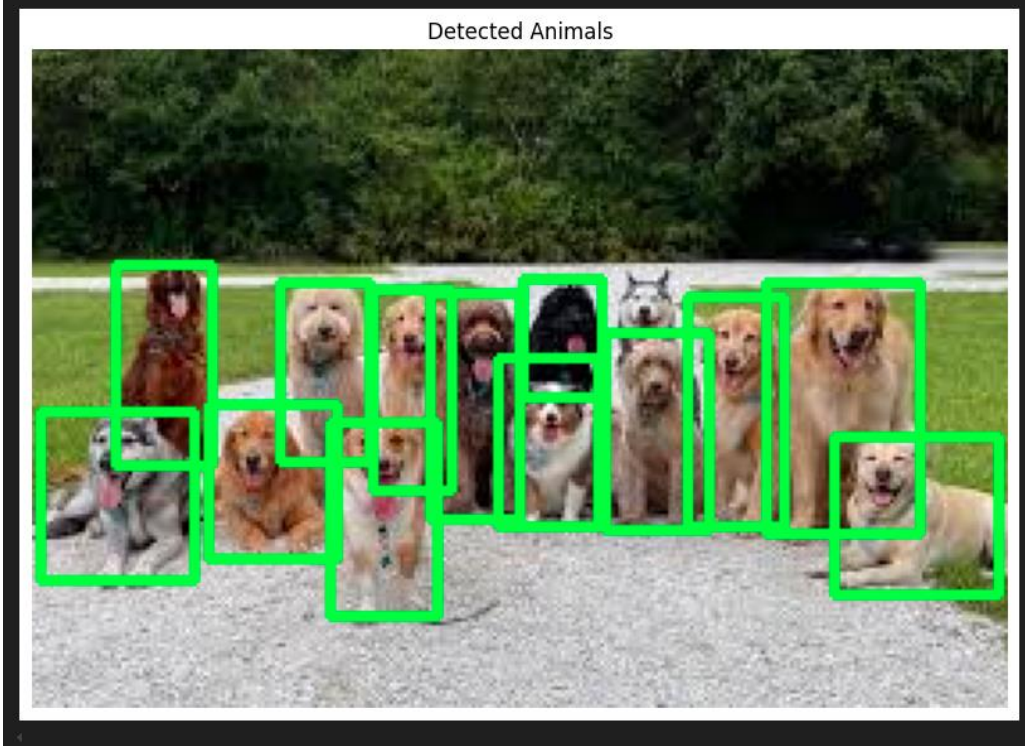
- Draws bounding boxes around detected animals on the image.
- Converts the image from BGR to RGB for proper visualization with Matplotlib.
- Displays the processed image with bounding boxes.
- Creates an interactive map with Folium:

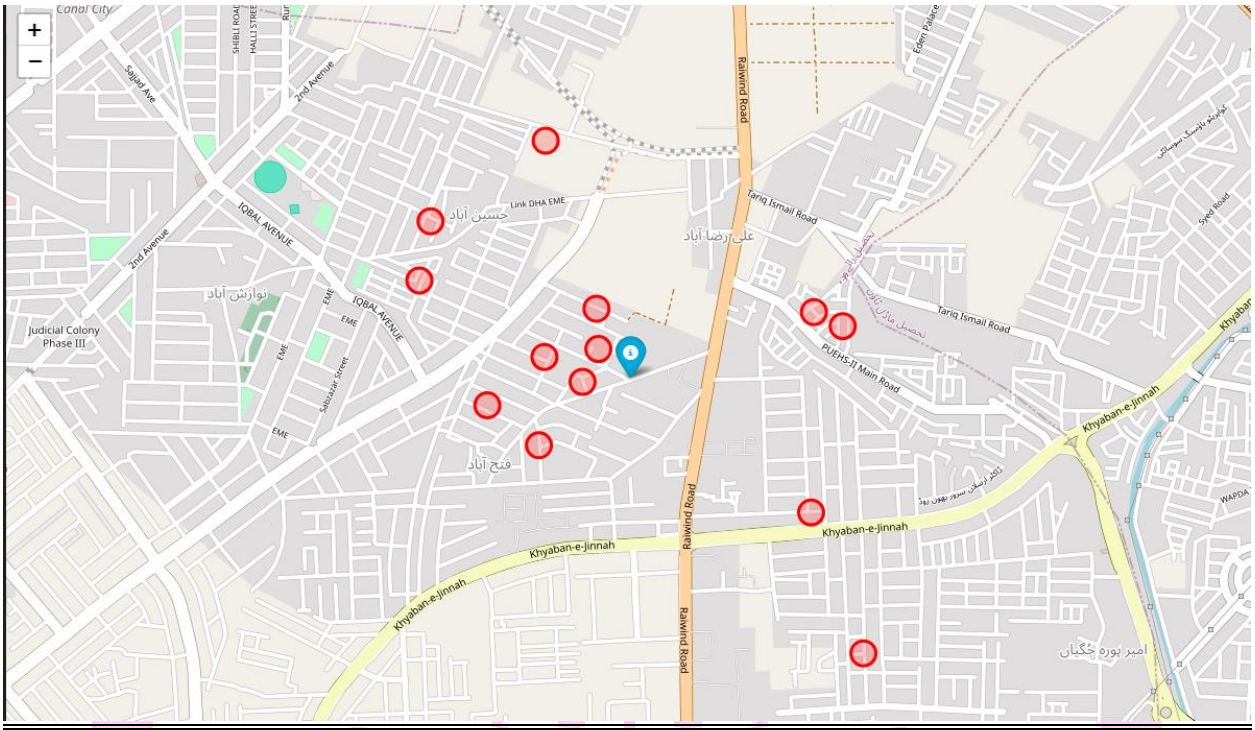
- Marks the user's current location.
- Places circles on the map to indicate the detected herd locations.

- Displays the interactive map for further exploration

OUTPUT

```
Alert! Animal herd detected nearbysuccesfully: Distance: 295.65 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 881.92 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 759.32 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 975.63 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 988.16 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 906.82 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 1410.10 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 447.21 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 189.16 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 345.59 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 165.60 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 852.07 meters
Alert! Animal herd detected nearbysuccesfully: Distance: 571.28 meters
```




Canal City
Sajjad Ave
2nd Avenue
IQBAL AVENUE
Judicial Colony Phase III
Salahuddin Street
Link DHA EME
Raiwind Road
Tariq Ismail Road
PUEHS-II Main Road
Khyaban-e-Jinnah
WAPDA
Syed Road