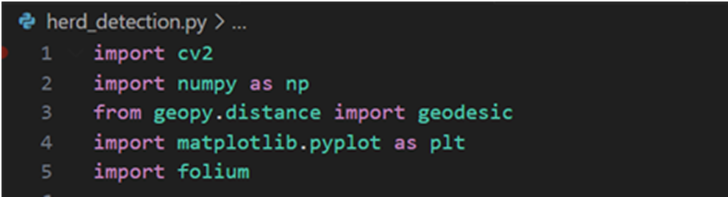
Name: M Shan Iftikhar

Roll No: BSAI-4A-017

Lab task 6

1. **Importing Necessary Libraries**



cv2: OpenCV is used for image processing and deep learning-based object detection.

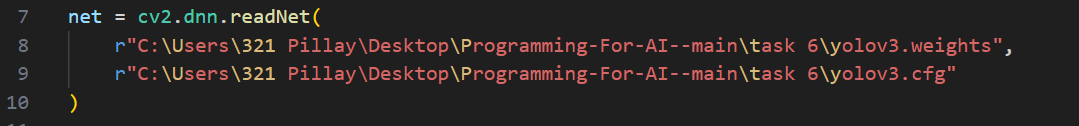
numpy: A numerical computing library used for mathematical operations and array manipulations.

geopy.distance: Helps calculate the geographical distance between two locations.

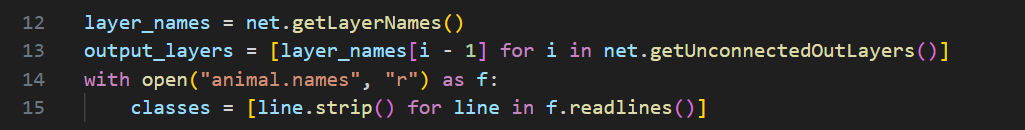
matplotlib.pyplot: Used for visualizing images with bounding boxes.

folium: Used for generating interactive maps to display herd locations.

1. **Loading the YOLOv3 Model**

****

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



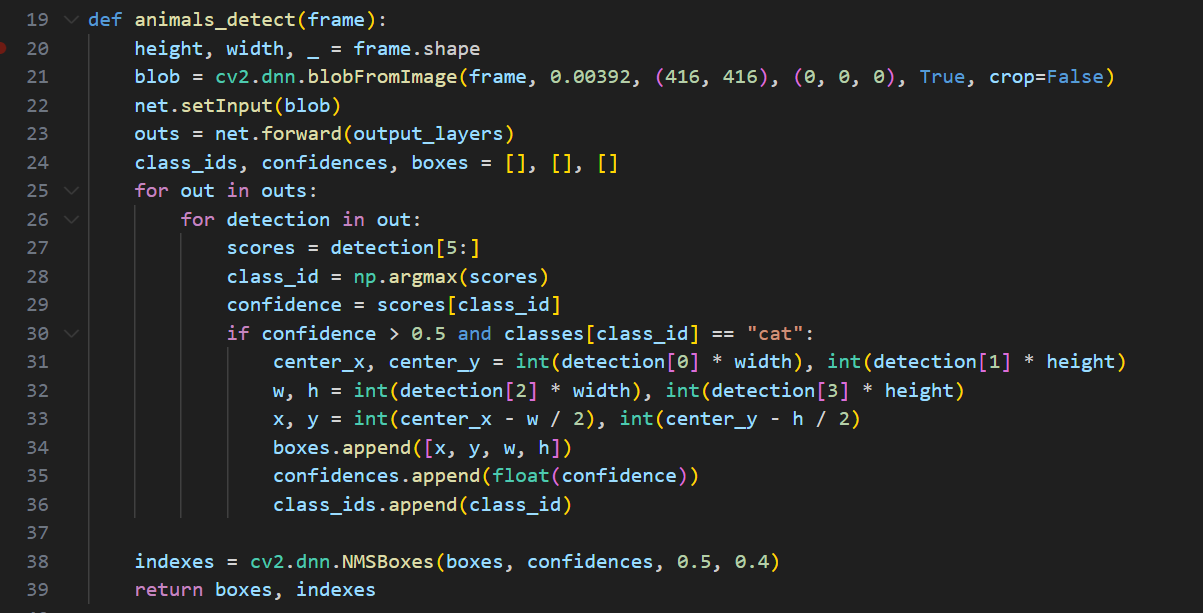
Retrieves the names of all layers in the YOLOv3 network.

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.

1. **Function to Detect Animals**

****

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().

Iterates through detected objects.

Extracts confidence scores and determines the class with the highest confidence.

Filters out low-confidence detections and only processes the "cat" class

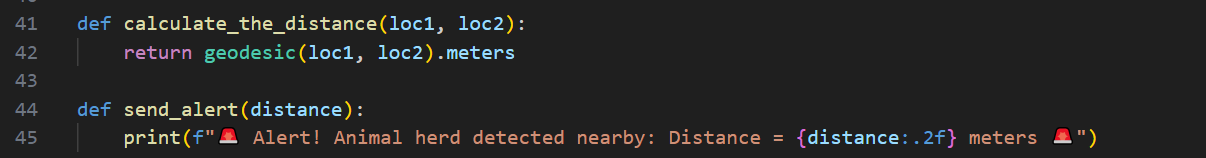
Converts YOLOv3 normalized coordinates to actual pixel values.

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.

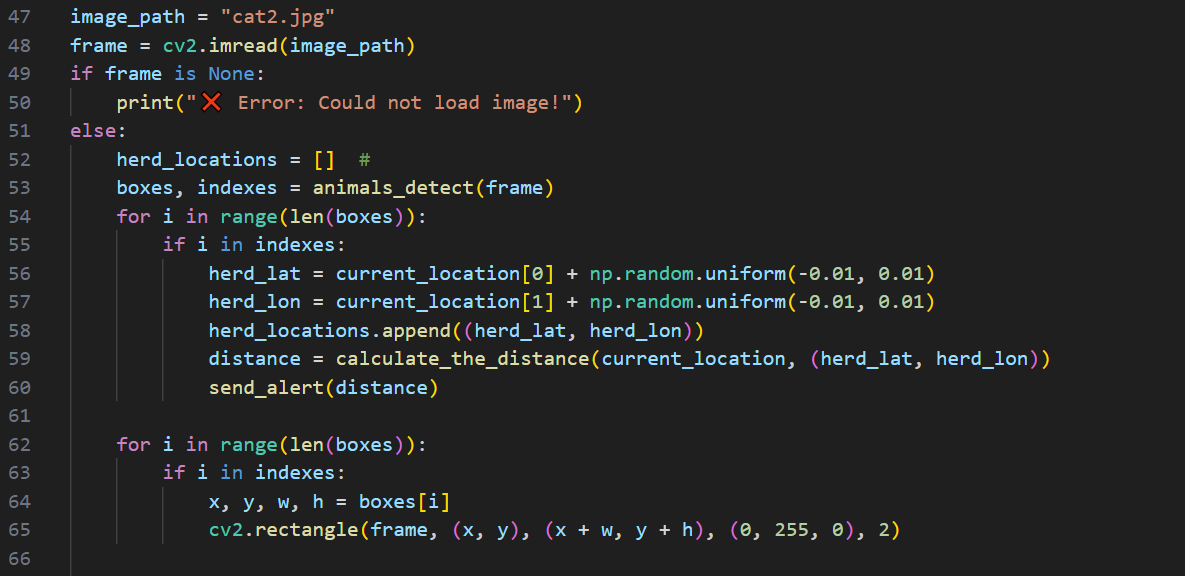
1. **Distance Calculation & Alerts**



geopy.distance.geodesic() calculates the real-world distance (in meters) between two geographical coordinates.

Prints an alert message when animals are detected near a specified location.

1. **Processing the Image for Detection**



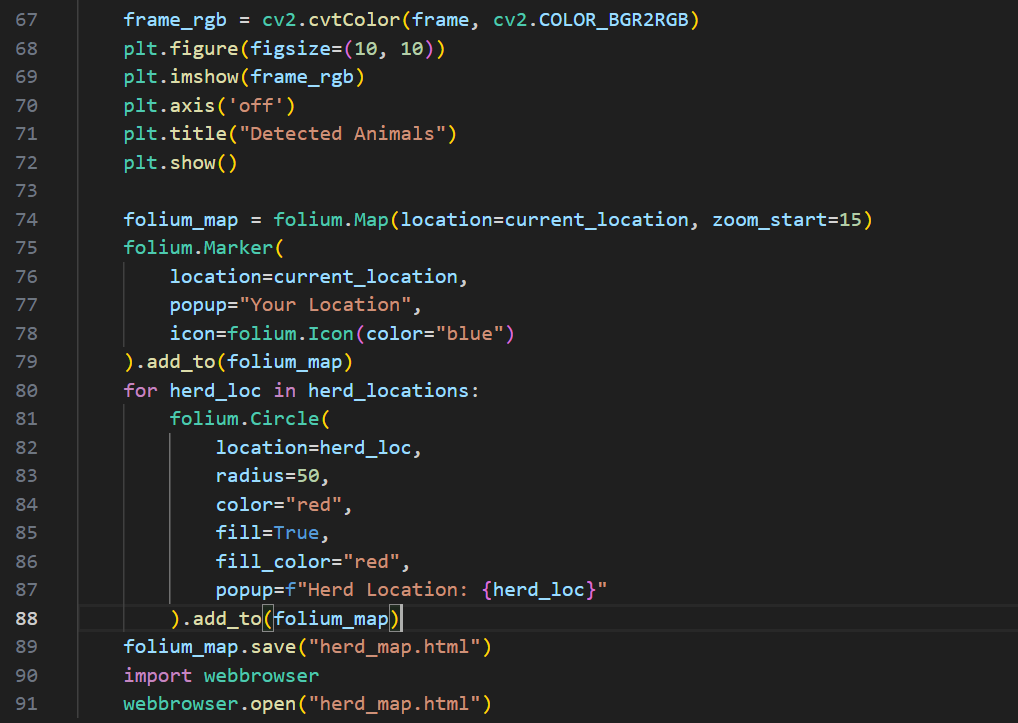
Loads an image (e.g., "cat1.jpg") and checks for successful loading.

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.

1. **Displaying Results**



Draws bounding boxes around detected animals on the image.

Converts the image from BGR to RGB for proper visualization with Matplotlib.

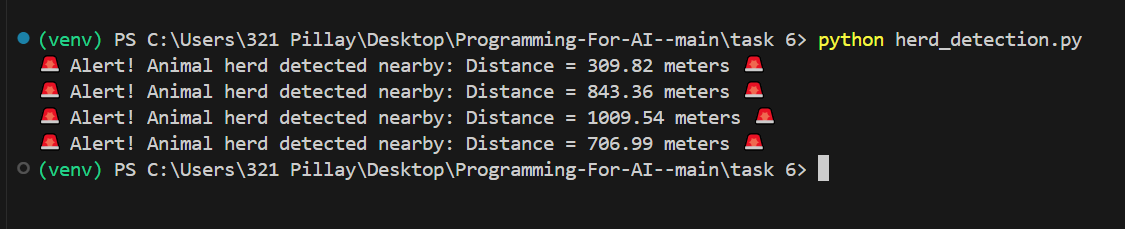
Displays the processed image with bounding boxes.

Creates an interactive map with Folium

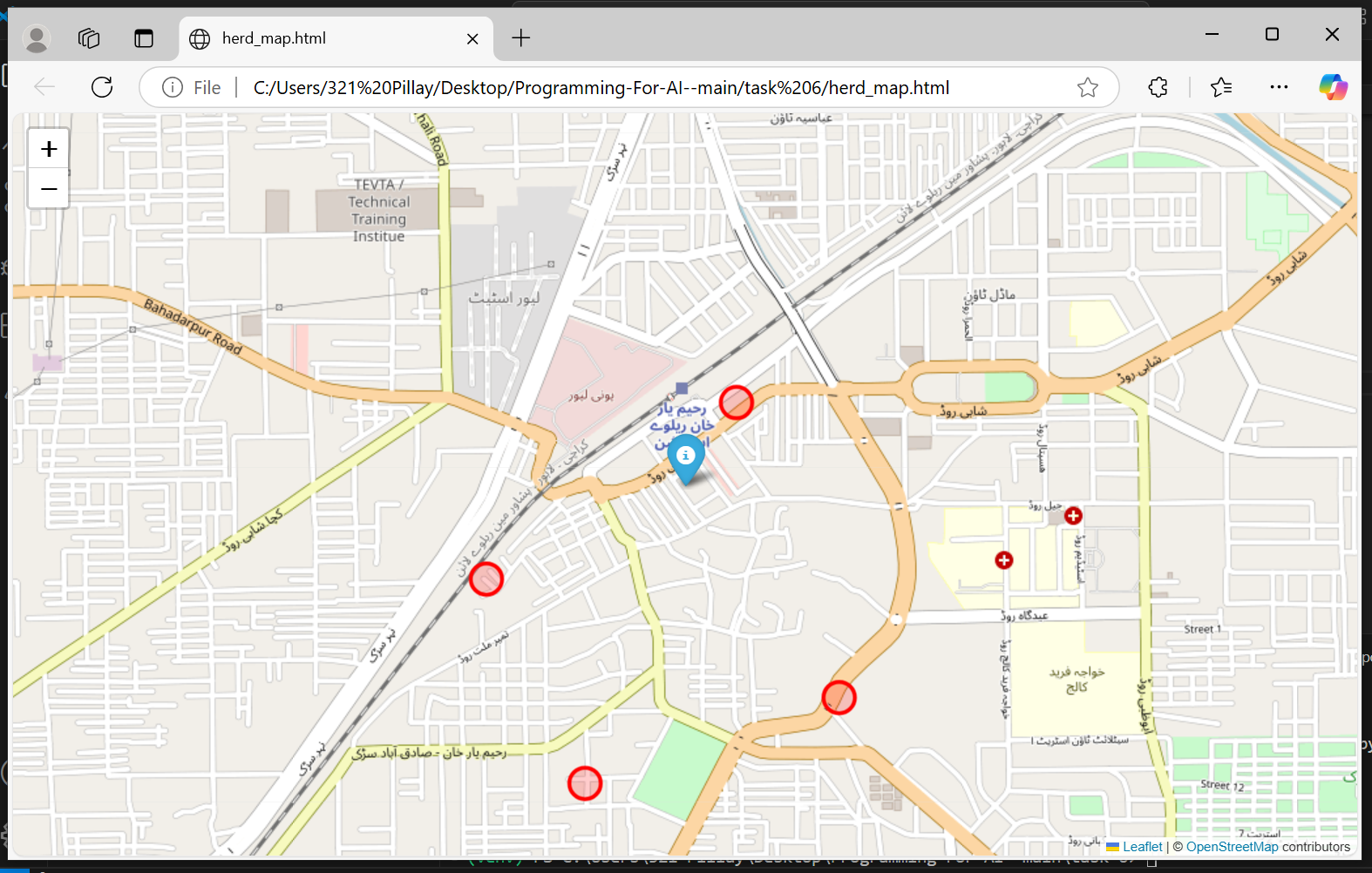
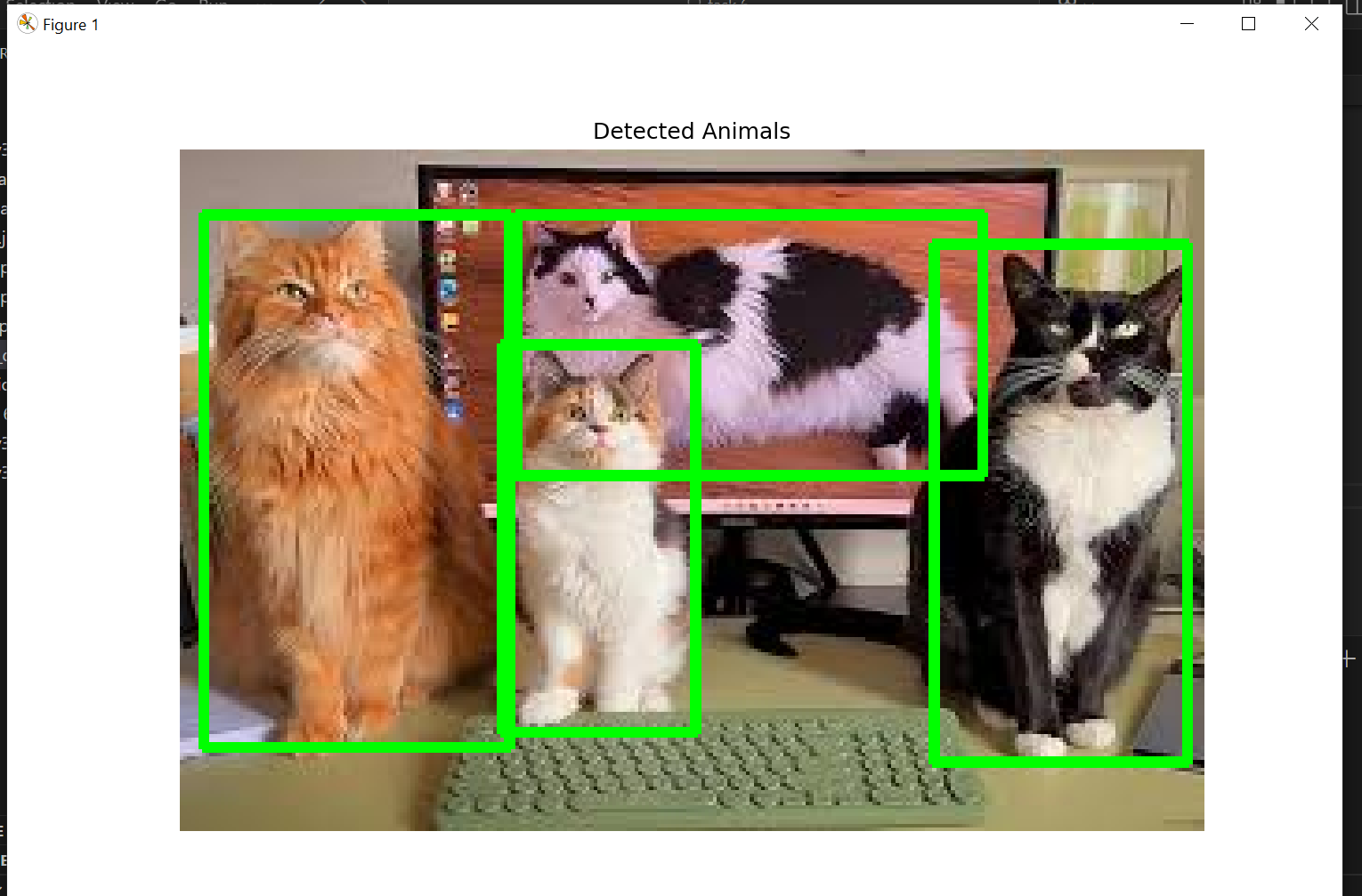
Marks the user’s current location.

Places circles on the map to indicate the detected herd locations.

Displays the interactive map for further exploration.

****

**Output**

****