**EXPLANATION FOR THE CODE:**

import cv2

import numpy as np

# Load YOLO model and its configuration

yolo\_net = cv2.dnn.readNet('yolov3.weights', 'yolov3.cfg')

classes = []

with open('coco.names', 'r') as f:

classes = f.read().strip().split('\n')

# Set the target for weed class (modify class\_id according to your YOLO model)

weed\_class = "weed"

class\_id = classes.index(weed\_class)

Explanation:

* We start by importing necessary libraries, including OpenCV (**cv2**) and NumPy (**numpy**).
* We load the YOLO model and its configuration files using **cv2.dnn.readNet**.
* We read the class names that the YOLO model is trained to detect from the file **coco.names** and store them in the **classes** list.
* We set the target class for weed detection by finding its index in the **classes** list.

def load\_dataset(dataset\_path):

# TODO: Load the dataset from the given dataset\_path and preprocess it

# You need to read the images and their corresponding labels from the dataset

# and return them as a list of tuples, e.g., [(image\_path, label), ...]

Pass

Explanation:

* The **load\_dataset** function is a placeholder for loading and preprocessing the dataset. It takes the path to the dataset directory (**dataset\_path**) as input.
* The function should be implemented by reading the images and their corresponding labels (crop or weed) from the dataset and returning them as a list of tuples. Each tuple contains the image path and its label, e.g., **[("image1.jpg", "crop"), ("image2.jpg", "weed"), ...]**.

def detect\_weeds(image):

# Prepare the image for YOLO

blob = cv2.dnn.blobFromImage(image, 1/255.0, (416, 416), swapRB=True, crop=False)

yolo\_net.setInput(blob)

# Run forward pass through the network

detections = yolo\_net.forward()

weed\_mask = np.zeros(image.shape[:2], dtype=np.uint8)

for detection in detections:

for obj in detection:

scores = obj[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.5: # You can adjust the confidence threshold

if classes[class\_id] == weed\_class:

x, y, w, h = (obj[0:4] \* np.array([image.shape[1], image.shape[0], image.shape[1], image.shape[0]])).astype(int)

weed\_mask[y:y+h, x:x+w] = 255

return weed\_mask

Explanation:

* The **detect\_weeds** function takes an image as input and performs weed detection using the YOLO model.
* The image is preprocessed for YOLO using **cv2.dnn.blobFromImage**.
* The YOLO model is then run on the preprocessed image, and the detections are obtained from **yolo\_net.forward()**.
* The function iterates through the detections and, if a weed is detected with sufficient confidence (0.5 in this case), it creates a binary weed mask (**weed\_mask**) by marking the detected weed region with white (255) pixels.

def spray\_weeds(weed\_mask):

for row in range(weed\_mask.shape[0]):

for col in range(weed\_mask.shape[1]):

if weed\_mask[row, col]:

print(f"Spraying weed at position ({col}, {row})")

Explanation:

* The **spray\_weeds** function simulates the spraying process by printing the positions of the detected weeds on the **weed\_mask**.
* The function iterates through the binary **weed\_mask**, and whenever it finds a white pixel (representing a detected weed region), it prints the position as "Spraying weed at position (x, y)".

def main():

dataset\_path = "path/to/your/dataset"

dataset = load\_dataset(dataset\_path)

for image\_path, label in dataset:

image = cv2.imread(image\_path)

weed\_mask = detect\_weeds(image)

spray\_weeds(weed\_mask)

# Run the main function

if \_\_name\_\_ == "\_\_main\_\_":

main()

Explanation:

* The **main** function is the entry point of the program.
* It sets the path to the dataset directory (**dataset\_path**) and calls the **load\_dataset** function to load and preprocess the dataset.
* For each image in the dataset, the **main** function reads the image, performs weed detection using **detect\_weeds**, and simulates the spraying process by calling **spray\_weeds**.

NOTE:. Remember to replace **"path/to/your/dataset"** with the actual path to your extracted dataset directory. Also, make sure that your dataset contains images of crops and weeds along with their corresponding labels (crop or weed). The code will then simulate the spraying process by printing the positions of the detected weeds for each image in the dataset.