

```

import cv2
import numpy as np
import datetime
import os
import time

CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
            "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
            "dog", "horse", "motorbike", "person", "pottedplant",
            "sheep", "sofa", "train", "tvmonitor"]

ANIMAL_CLASSES = {"bird", "cat", "cow", "dog", "horse", "sheep"}

net = cv2.dnn.readNetFromCaffe("MobileNetSSD_deploy.prototxt",
                               "MobileNetSSD_deploy.caffemodel")

os.makedirs("captures", exist_ok=True)

log_file_path = "animal_detections.txt"
open(log_file_path, "a").close()

cap = cv2.VideoCapture(0)

last_log_time = 0
DETECTION_DELAY = 2

print("Okay, wildlife monitor is running. Press 'q' when you're done.")

while True:
    ret, frame = cap.read()
    if not ret:
        print("Couldn't get a frame from the webcam. Quitting.")
        break

    h, w = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 0.007843,
                                  (300, 300), 127.5)

    net.setInput(blob)
    detections = net.forward()
    current_time = time.time()

    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > 0.5:
            idx = int(detections[0, 0, i, 1])
            label = CLASSES[idx]
            if label in ANIMAL_CLASSES:
                box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                (startX, startY, endX, endY) = box.astype("int")
                color = (0, 255, 0)
                cv2.rectangle(frame, (startX, startY), (endX, endY), color,
2)

                text = f"{label}: {confidence:.2f}"
                y = startY - 10 if startY - 10 > 10 else startY + 10
                cv2.putText(frame, text, (startX, y),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

                if current_time - last_log_time > DETECTION_DELAY:
                    timestamp = datetime.datetime.now().strftime("%Y-%m-
%d %H-%M-%S")

```

```
        filename = f"{label}_{timestamp}.jpg"
        filepath = os.path.join("captures", filename)
        cv2.imwrite(filepath, frame)

        with open(log_file_path, "a") as log_file:
            log_file.write(f"{timestamp} - {label.upper()}
spotted with {confidence:.2f} confidence\n")

        print(f"{label.upper()} spotted at {timestamp}")
        last_log_time = current_time

cv2.imshow("Wildlife Monitoring", frame)

if cv2.waitKey(1) & 0xFF == ord('q'):
    print("Stopping the monitor. Goodbye!")
    break

cap.release()
cv2.destroyAllWindows()
```