**AI POWERED SURVEILLANCE**

import cv2

import numpy as np

import torch

import os

import requests

from torchvision.models.detection import fasterrcnn\_resnet50\_fpn, FasterRCNN\_ResNet50\_FPN\_Weights

# Download required YOLO files if they don't exist

def download\_yolo\_files():

    yolo\_files = {

        'yolov3.weights': 'https://pjreddie.com/media/files/yolov3.weights',

        'yolov3.cfg': 'https://raw.githubusercontent.com/pjreddie/darknet/master/cfg/yolov3.cfg',

        'coco.names': 'https://raw.githubusercontent.com/pjreddie/darknet/master/data/coco.names'

    }

    for filename, url in yolo\_files.items():

        if not os.path.exists(filename):

            print(f"Downloading {filename}...")

            response = requests.get(url, stream=True)

            with open(filename, 'wb') as f:

                for chunk in response.iter\_content(chunk\_size=8192):

                    f.write(chunk)

# Load models

def load\_models():

    download\_yolo\_files()

    # Load YOLO model

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

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

        yolo\_classes = [line.strip() for line in f.readlines()]

    return yolo\_net, yolo\_classes

# Process frame with YOLO

def process\_yolo(frame, net, classes):

    height, width = frame.shape[:2]

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

    net.setInput(blob)

    output\_layers = net.getUnconnectedOutLayersNames()

    layer\_outputs = net.forward(output\_layers)

    weapons = {"knife", "gun", "pistol", "rifle"}

    for output in layer\_outputs:

        for detection in output:

            scores = detection[5:]

            class\_id = np.argmax(scores)

            confidence = scores[class\_id]

            if confidence > 0.5 and classes[class\_id] in weapons:

                print(f"⚠️ {classes[class\_id]} detected as an unusual object!")

                return True

    return False

# Open camera and detect objects

def main():

    yolo\_net, yolo\_classes = load\_models()

    cap = cv2.VideoCapture(0)

    if not cap.isOpened():

        print("Error: Could not open camera.")

        return

    while True:

        ret, frame = cap.read()

        if not ret:

            break

        detected = process\_yolo(frame, yolo\_net, yolo\_classes)

        if detected:

            cv2.putText(frame, "Warning: Weapon Detected!", (50, 50), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 0, 255), 2)

        cv2.imshow("Live Detection", frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

    cap.release()

    cv2.destroyAllWindows()

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

    main()