

Object Recognition

Overview of the Project:

Step 1: Get ready to discover amazing things!

We need special tools to make our computer super smart. Let's install them using a magical command called "pip install". Say, "Yay, magic tools!"

Step 2: Get your camera ready!

We want our computer to see things, just like our eyes do. We're going to use the camera to show the computer what's happening. Get your camera, and let's create a special camera object called "video"!

Step 3: Let's find hidden objects!

Imagine being a detective on a thrilling mission. We'll start a mission called "Operation Object Detection".

Our computer will take pictures and use its detective skills to find objects. It will even draw boxes around the objects it recognizes. How cool is that? Woohoo!

Step 4: Collecting our treasure list!

Every time our computer finds a new object, we'll add it to our treasure list. Imagine it like collecting precious gems or secret treasures!

We don't want the same object on our list twice. We only add new and exciting objects. Our treasure list keeps growing with unique finds!

Step 5: Let's make the computer talk!

Our computer has a hidden talent—it can talk just like a superstar! We'll make it say the names of the objects it found in a cool and exciting way.

We'll create a special magic function that turns text into speech. The computer will use its voice to speak to us. Get ready to hear some awesome announcements!

Code:

```
import cv2
import cvlib as cv
from cvlib.object_detection import draw_bbox
from gtts import gTTS
from playsound import playsound
from food_facts import food_facts

def speech(text):
    print(text)
    language = "en"
    output = gTTS(text=text, lang=language, slow=False)
    output.save("./sounds/output.mp3")
    playsound("./sounds/output.mp3")

video = cv2.VideoCapture(0)
labels = []
while True:
    ret, frame = video.read()
    bbox, label, conf = cv.detect_common_objects(frame)
    output_image = draw_bbox(frame, bbox, label, conf)
    cv2.imshow("Detection", output_image)
    for item in label:
        if item in labels:
            pass
        else:
            labels.append(item)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

i = 0
new_sentence = []
for label in labels:
    if i == 0:
        new_sentence.append(f"I found a {label}, and, ")
    else:
        new_sentence.append(f"a {label},")
    i += 1
speech(" ".join(new_sentence))
```

Explanation:

Step 1: Importing necessary libraries 📖

We start by importing some special libraries that will help us do cool things with our computer's camera, speech, and object detection.

These libraries are like toolboxes filled with magical functions and powers.

It Includes OpenCV,cvlib,gtts,playsound.

Step 2: Defining a speech function 🗣️

We create a special function called "speech" that will make our computer speak out loud.

This function takes a text as input and performs some magic to convert it into speech.

It saves the speech as an audio file and then plays it for us to hear using speakers or headphones.

Step 3: Setting up the camera 📷

We connect our computer's camera to the code using the "cv2.VideoCapture(0)" command.

This lets us see what the camera sees and capture frames (pictures) from it.

Step 4: Preparing for object detection 🕵️♀️🔍

We create an empty list called "labels" to store the names of the objects we detect.

Step 5: Starting the object detection loop 🔁

We enter a loop that will keep running until we decide to stop.

Inside the loop, we read frames (pictures) from the camera using `"video.read()"`.

We also use a special function called `"cv.detect_common_objects"` to detect objects in each frame.

The detected objects are returned as bounding boxes (boxes around the objects), labels (names of the objects), and confidence scores.

Step 6: Displaying the object detection results

We draw bounding boxes around the detected objects on the frame using `"draw_bbox"` function.

This helps us visualize where the objects are located in the image.

The resulting image with bounding boxes is displayed in a window called `"Detection"`.

Step 7: Collecting unique object labels

We go through each label (object name) detected in the current frame.

If the label is already in our `"labels"` list, we skip it because we only want unique labels.

Otherwise, we add the label to our `"labels"` list.

Step 8: Exiting the loop

We check if the user has pressed the `"q"` key on the keyboard using `"cv2.waitKey(1) & 0xFF == ord('q')"`.

If the `"q"` key is pressed, we break out of the loop and stop the program.

Step 9: Creating a sentence with the object labels

We create an empty list called `"new_sentence"` to store parts of a sentence.

We go through each label in the "labels" list and add them to the sentence with some fun phrases. The sentence describes the objects we found using their labels.

Step 10: Making the computer speak the sentence 🗣️🔊

We call the "speech" function we defined earlier and pass the sentence as text. The computer performs magic to convert the text into speech and speaks it out loud.

OUTPUT:



