Nama : Rosdan

Nim : 230741099

Makul: Kecerdasan Tiruan

Penjelasan Object Detection Projrct

1.Pendahuluan

Kode ini adalah aplikasi deteksi objek real-time menggunakan model YOLO (You Only Look Once) yang diintegrasikan dengan Streamlit untuk antarmuka web. Aplikasi ini memanfaatkan kamera untuk mendeteksi objek dalam waktu nyata dan menampilkan hasil deteksi berupa bounding box dan label objek.

2.Penjelasan kode

a. Impor Library

```
from ultralytics import YOLO
import cv2
import streamlit as st
from PIL import Image
import numpy as np
from collections import Counter
```

Kode ini mengimpor berbagai library:

YOLO: Untuk memuat model YOLO dan melakukan deteksi objek.

cv2 (OpenCV): Untuk pengolahan citra, seperti membaca video feed dan menggambar bounding box.

streamlit: Untuk membuat antarmuka aplikasi berbasis web.

PIL.Image: Untuk memproses gambar.

numpy: Untuk manipulasi array.

Counter: Untuk menghitung jumlah setiap jenis objek yang terdeteksi.

b. Fungsi load_model

```
@st.cache_resource
def load_model(model_path):
return YOLO(model_path)
```

Deskripsi: Fungsi ini memuat model YOLO dari path yang diberikan.

Parameter:

model_path: Path file model YOLO (contoh: yolo11n.pt).

Decorator:

@st.cache_resource: Menyimpan model dalam cache agar tidak dimuat ulang setiap kali aplikasi di-refresh, sehingga menghemat waktu.

Return Value: Objek model YOLO yang telah dimuat.

c. Fungsi display_results

```
def display_results(image, results):
    ...
  return image, detected_objects
```

Deskripsi:

Fungsi ini bertanggung jawab untuk memproses hasil deteksi dari model YOLO dan menggambar bounding box pada gambar input.

Parameter:

1. image: Gambar input (frame dari kamera).

2. results: Hasil deteksi objek dari YOLO.

Parameter:

1. image: Gambar input (frame dari kamera).

2. results: Hasil deteksi objek dari YOLO.

Proses:

1. Ekstraksi Informasi:

boxes: Koordinat bounding box ([x1, y1, x2, y2]).

scores: Confidence score untuk setiap deteksi.

labels: Indeks kelas dari objek yang terdeteksi.

names: Nama kelas objek.

2.Filter Berdasarkan Confidence:

Deteksi hanya diproses jika confidence score > 0.5.

3. Menggambar Bounding Box:

Kotak hijau (cv2.rectangle) digambar di sekitar objek.

Label dan skor confidence ditambahkan di atas bounding box (cv2.putText).

4. Menyimpan Deteksi:

Nama objek yang terdeteksi disimpan dalam list detected_objects.

Output:

image: Gambar dengan bounding box dan label.

detected_objects: List objek yang terdeteksi.

d. Fungsi Utama main

def main():

...

Deskripsi:

Fungsi utama yang mengatur alur aplikasi, termasuk:

Memuat model.

Mengaktifkan kamera.

Menampilkan hasil deteksi pada antarmuka Streamlit.

Langkah-Langkah:

1. Menampilkan Judul Aplikasi:

st.title("Real-time Object Detection with YOLO")

st.sidebar.title("Settings")

2. Memuat Model YOLO:

model_path = "yolo11n.pt"

model = load_model(model_path)

Model YOLO dimuat menggunakan fungsi load_model.

3. Checkbox untuk Mengontrol Deteksi:

run_detection = st.sidebar.checkbox("Start/Stop Object Detection", key="detection_control")

Checkbox ini memungkinkan pengguna untuk memulai atau menghentikan deteksi objek.

4. Membuka Kamera:

```
if run_detection:
    cap = cv2.VideoCapture(0)
    st_frame = st.empty()  # Placeholder untuk video feed
    st_detection_info = st.empty()  # Placeholder untuk informasi deteksi
```

Kamera dibuka menggunakan OpenCV (cv2.VideoCapture(0)).

Placeholder dibuat untuk menampilkan video dan informasi deteksi.

5. Loop Deteksi:

```
while True:
    ret, frame = cap.read()
    ...
```

Langkah-langkah:

1. Membaca Frame Kamera:

Jika gagal membaca, tampilkan peringatan.

2. Konversi Warna:

```
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

Frame dikonversi dari BGR ke RGB untuk ditampilkan di Streamlit.

3. Prediksi Objek:

```
results = model.predict(frame, imgsz=640)
```

YOLO mendeteksi objek pada frame dengan ukuran gambar 640x640.

4. Proses Hasil Deteksi:

```
frame, detected_objects = display_results(frame, results[0])
```

Bounding box dan label digambar pada frame.

5. Tampilkan Video dan Informasi Deteksi:

```
st_frame.image(frame, channels="RGB", use_column_width=True)
st_detection_info.text(detection_info)
```

Video ditampilkan di Streamlit.

Informasi jumlah objek yang terdeteksi (menggunakan Counter) juga ditampilkan.

6. Periksa Status Checkbox:

if not st.session_state.detection_control:

break

Loop dihentikan jika checkbox dinonaktifkan.

7.Melepaskan kamera

Cap.realease()

Kamera dilepaskan setelah selesai digunakan

e. Menjalankan Aplikasi

if _name_ == "_main_":

main()

Memastikan bahwa fungsi main dijalankan hanya jika skrip dijalankan secara langsung, bukan diimpor sebagai modul.