

Nama : Muhammad Qodrat Hanif Fadhila

Nim : 230741110

Mata Kuliah : UAS Kecerdasan Tiruan

Program Studi : Ilmu Komputer

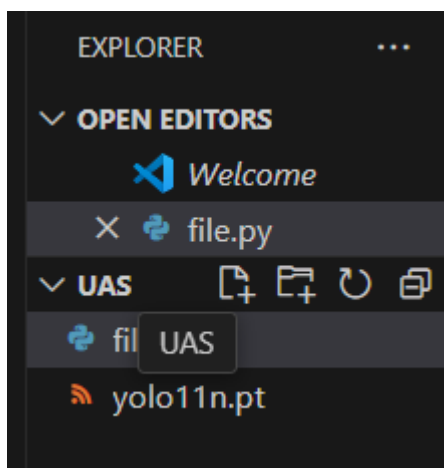
Tutorial Membuat Aplikasi Deteksi Objek Menggunakan YOLO dan Streamlit

Pertama

Buat folder terlebih dahulu dengan nama yang kalian inginkan

Setelah itu buka VS Code dan buka folder yang telah kita buat

Kemudian kita buat didalamnya file app.py



setelah itu masukan Model YOLO yang Anda gunakan untuk deteksi objek di folder Object Detection Project

Setelah itu kita buat file app.py, kita masukan kode pemrograman seperti dibawah ini

```
1  from ultralytics import YOLO
2  import cv2
3  import streamlit as st
4  from PIL import Image
5  import numpy as np
6  from collections import Counter
7
8  # Load YOLO model
9  @st.cache_resource
10 def load_model(model_path):
11     return YOLO(model_path)
12
13 # Process and display the detection results
14 def display_results(image, results):
15     boxes = results.boxes.xyxy.cpu().numpy() # [x1, y1, x2, y2]
16     scores = results.boxes.conf.cpu().numpy() # Confidence scores
17     labels = results.boxes.cls.cpu().numpy() # Class indices
18     names = results.names # Class names
19
20     detected_objects = []
21
22     for i in range(len(boxes)):
23         if scores[i] > 0.5: # Confidence threshold
24             x1, y1, x2, y2 = boxes[i].astype(int)
25             label = names[int(labels[i])]
26             score = scores[i]
27             detected_objects.append(label)
28             cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
29             cv2.putText(image, f"{label}: {score:.2f}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
30
31     return image, detected_objects
32
33 # Main Streamlit app
34 def main():
35     st.title("MENUJU TAK TERBATAS DAN MEPLAMPAUINYA Rohman")
36     st.sidebar.title("Settings")
37
38     model_path = "yolo11n.pt" # Path to your YOLO model
39     model = load_model(model_path)
40
41     # Create the checkbox once
42     run_detection = st.sidebar.checkbox("Start/Stop Object Detection", key="detection_control")
43
44     # Open video capture if checkbox is active
45     if run_detection:
46         cap = cv2.VideoCapture(0)
47         st_frame = st.empty() # Placeholder for video frames
48         st_detection_info = st.empty() # Placeholder for detection information
49
50         while True:
51             ret, frame = cap.read()
52             if not ret:
53                 st.warning("Failed to capture image.")
54                 break
55
56             # Run YOLO detection
57             frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert to RGB for display
58             results = model.predict(frame, imgsz=640) # Perform detection
59
60             # Draw results and collect detected objects
61             frame, detected_objects = display_results(frame, results[0])
62
63             # Display video feed
64             st_frame.image(frame, channels="RGB", use_column_width=True)
65
66             # Display detection information
67             if detected_objects:
68                 object_counts = Counter(detected_objects)
69                 detection_info = "\n".join([f"{obj}: {count}" for obj, count in object_counts.items()])
70             else:
71                 detection_info = "No objects detected."
72
```

```
72
73     st_detection_info.text(detection_info) # Update detection info text
74
75     # Break the loop if checkbox is unchecked
76     if not st.session_state.detection_control:
77         break
78
79     cap.release()
80
81 if __name__ == "__main__":
82     main()
```

Kemudian kita install library di terminal kita :

```
PS C:\Users\rois2\UAS> pip install streamlit
PS C:\Users\rois2\UAS> pip install ultralytics
```

Dan setelah kita lakukan semua diatas kita dapat menjalankannya dengan kode berikut:

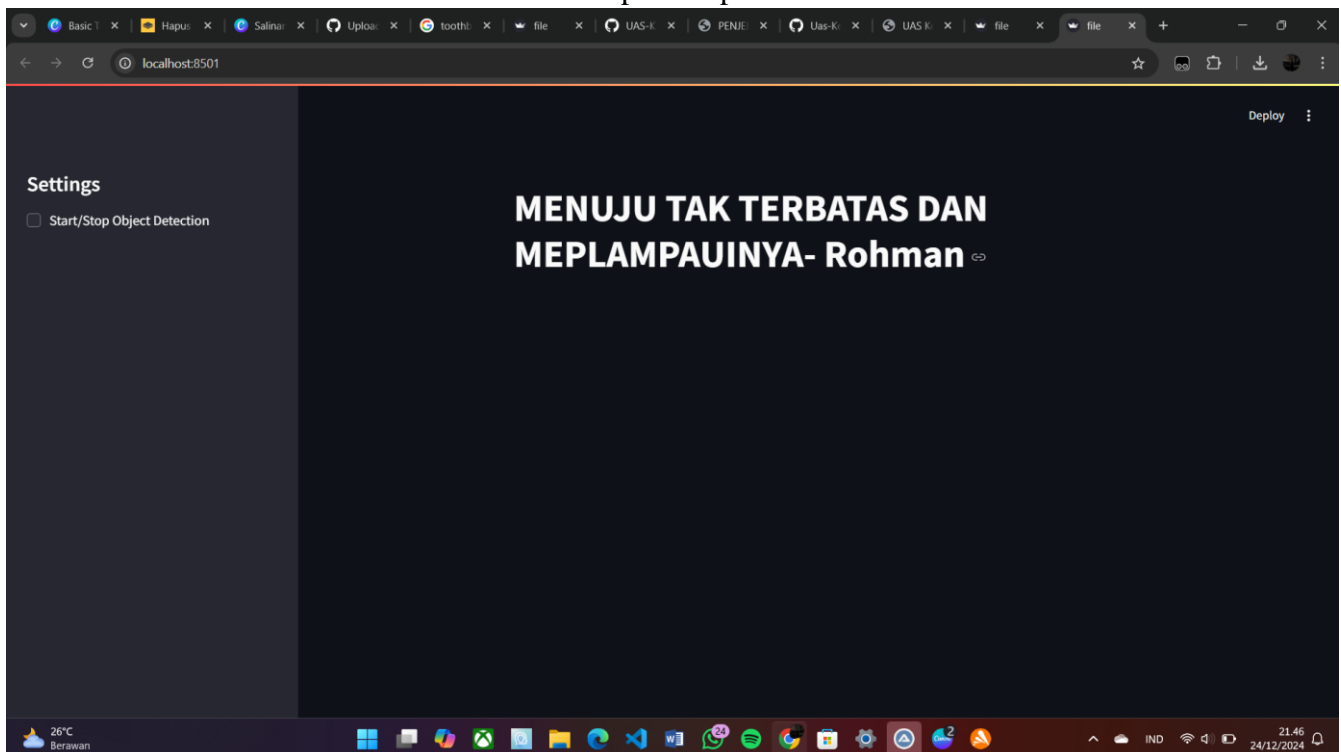
```
PS C:\Users\rois2\UAS> streamlit run file.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.43.53:8501

2024-12-24 21:49:13.076 Examining the path of torch.classes raised: Tried via torch::class_
```

Setelah menjalankan 'streamlit run app.py, nanti akan muncul perintah isi email, kalian tinggal isi email kalian kemudian akan muncul tampilan seperti berikut



Apabila kita tekan tombol mulai di sidebar akan memulai deteksi



itulah tutorial singkat dari saya semoga mudah dipahami dan selamat mencoba

PENJELASAN TENTANG KODE PEMOGRAMAN

1. Impor Library

```
from ultralytics import YOLO
import cv2
import streamlit as st
from PIL import Image
import numpy as np
from collections import Counter
import base64
```

- 1) **YOLO (Ultralytics):** Untuk memuat model YOLO yang digunakan dalam deteksi objek.

- 2) **OpenCV (cv2)**: Digunakan untuk pengolahan gambar seperti menggambar kotak pembatas (bounding box) dan membaca input dari kamera.
- 3) **Streamlit (st)**: Membuat antarmuka aplikasi web.
- 4) **Pillow (PIL)**: Digunakan untuk manipulasi gambar.
- 5) **NumPy (np)**: Operasi array untuk memproses hasil deteksi.
- 6) **Counter**: Menghitung jumlah objek yang terdeteksi.
- 7) **Base64**: Mengonversi file (gambar atau GIF) menjadi format yang dapat disematkan di aplikasi web.

2. Fungsi untuk Memuat Model YOLO

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

- **Tujuan:** Memuat model YOLO dari file .pt.
- **Parameter:**
 - model_path: Path ke file model YOLO, misalnya yolo11n.pt.
 - @st.cache_resource: Streamlit menyimpan (caching) model yang telah dimuat untuk mempercepat proses jika fungsi dipanggil kembali.

5. Fungsi untuk Memproses dan Menampilkan Hasil Deteksi

```
# Process and display the detection results
def display_results(image, results):
    boxes = results.boxes.xyxy.cpu().numpy() # [x1, y1, x2, y2]
    scores = results.boxes.conf.cpu().numpy() # Confidence scores
    labels = results.boxes.cls.cpu().numpy() # Class indices
    names = results.names # Class names

    detected_objects = []

    for i in range(len(boxes)):
        if scores[i] > 0.5: # Confidence threshold
            x1, y1, x2, y2 = boxes[i].astype(int)
            label = names[int(labels[i])]
            score = scores[i]
            detected_objects.append(label)
            cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.putText(image, f"{label}: {score:.2f}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    return image, detected_objects
```

- **Tujuan:**
 1. Memproses hasil deteksi objek dari YOLO.
 2. Menambahkan kotak pembatas (bounding box) dan label pada gambar.
 3. Mengembalikan gambar yang telah diproses dan daftar objek yang terdeteksi.
- **Parameter:**
 - a. image: Gambar input (dari kamera).

b. results: Hasil deteksi dari model YOLO.

- **Cara Kerja:**

1. **Koordinat Kotak:** Mengambil koordinat [x1, y1, x2, y2] untuk kotak pembatas setiap objek.
2. **Confidence Score:** Mengecek nilai kepercayaan (confidence) untuk memfilter deteksi yang tidak relevan.
3. **Label Objek:** Menambahkan nama objek di atas kotak pembatas.
4. **OpenCV:** Menggambar kotak dan label pada gambar.

6. Fungsi Utama

```
# Main Streamlit app
def main():
    st.title("MENUJU TAK TERBATAS DAN MEPLAMPAUINYA- Rohman")
    st.sidebar.title("Settings")

    model_path = "yolo11n.pt" # Path to your YOLO model
    model = load_model(model_path)

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

    # Open video capture if checkbox is active
    if run_detection:
        cap = cv2.VideoCapture(0)
        st_frame = st.empty() # Placeholder for video frames
        st_detection_info = st.empty() # Placeholder for detection information

        while True:
            ret, frame = cap.read()
            if not ret:
                st.warning("Failed to capture image.")
                break

            # Run YOLO detection
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert to RGB for display
            results = model.predict(frame, imgsz=640) # Perform detection

            # Draw results and collect detected objects
            frame, detected_objects = display_results(frame, results[0])

            # Display video feed
            st_frame.image(frame, channels="RGB", use_column_width=True)
```

```
# Display detection information
if detected_objects:
    object_counts = Counter(detected_objects)
    detection_info = "\n".join([f"{obj}: {count}" for obj, count in object_counts.items()])
else:
    detection_info = "No objects detected."

st_detection_info.text(detection_info) # Update detection info text

# Break the loop if checkbox is unchecked
if not st.session_state.detection_control:
    break

cap.release()
```

- **Fungsi:**

1. Menampilkan antarmuka aplikasi (judul, menu, latar belakang).
2. Memulai dan menghentikan proses deteksi objek.
3. Mengakses kamera untuk mendeteksi objek secara langsung.
4. Menampilkan hasil deteksi dan informasi objek di aplikasi.

- **Bagian Penting:**

- **Judul Aplikasi:** Ditampilkan di bagian atas aplikasi menggunakan st.title.
- **GIF Sidebar:** Menampilkan animasi di sidebar untuk estetika.
- **Kontrol Deteksi:**
 - ✦ Tombol "Mulai" mengaktifkan deteksi objek.
 - ✦ Tombol "Berhenti" menghentikan deteksi.
- **Video Streaming:**
 - ✦ Menggunakan cv2.VideoCapture(0) untuk mengakses kamera.
 - ✦ Proses gambar frame demi frame untuk deteksi objek.

7. Menjalankan Program

```
if __name__ == "__main__":  
    main()
```

Fungsi: Memastikan aplikasi berjalan ketika file dieksekusi langsung.