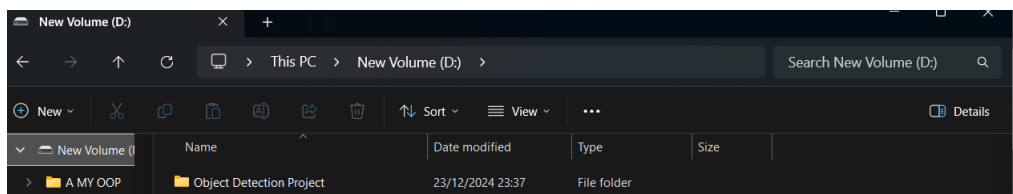


Nama : Muhammad Qodrat Hanif Fadhila
Nim : 230741110
Mata Kuliah : UAS Kecerdasan Tiruan
Program Studi : Ilmu Komputer

TUTORIAL MEMBUAT APLIKASI DETEKSI OBJEK MENGUNAKAN YOLO DAN STREAMLIT

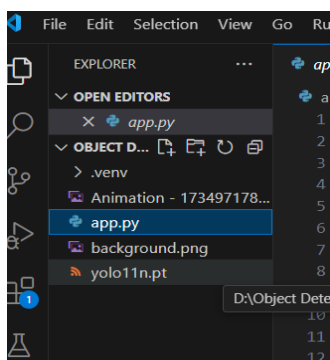
Pertama

Buat folder terlebih dahulu dengan nama “Object Detection Project”



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

Kemudian kita buat didalamnya file app.py



Jangan lupa kita buat venv bisa dibuat di terminal atau di cmd dengan kode:

```
python -m venv nama_venv
```

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

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

```
app.py x
app.py > ...
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 import base64 # Untuk encoding gambar ke base64
8
9 # Fungsi untuk menambahkan latar belakang
10 def set_background(image_path):
11     with open(image_path, "rb") as file:
12         base64_image = base64.b64encode(file.read()).decode()
13     css = f"""
14     <style>
15     .stApp {{
16         background-image: url("data:image;base64,{base64_image}");
17         background-size: cover;
18         background-repeat: no-repeat;
19         background-attachment: fixed;
20     }}
21     </style>
22     """
23     st.markdown(css, unsafe_allow_html=True)
24
25 # Fungsi untuk menampilkan GIF di sidebar
26 def display_animation(gif_path):
27     with open(gif_path, "rb") as file:
28         gif_data = file.read()
29     base64_gif = base64.b64encode(gif_data).decode()
30     st.sidebar.markdown(
31         f"""
32         <div style="text-align: center;">
33             
34         </div>
35         """
36         ,
37         unsafe_allow_html=True,
38     )
39
40 # Load YOLO model
41 @st.cache_resource
42 def load_model(model_path):
43     return YOLO(model_path)
44
45 # Function untuk memproses hasil deteksi
46 def display_results(image, results):
47     boxes = results.boxes.xyxy.cpu().numpy() # [x1, y1, x2, y2]
48     scores = results.boxes.conf.cpu().numpy() # Confidence scores
49     labels = results.boxes.cls.cpu().numpy() # Class indices
50     names = results.names # Class names
51
52     detected_objects = []
53
54     for i in range(len(boxes)):
55         if scores[i] > 0.5: # Confidence threshold
56             x1, y1, x2, y2 = boxes[i].astype(int)
57             label = names[int(labels[i])]
58             score = scores[i]
59             detected_objects.append(label)
60             cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
61             cv2.putText(image, f"{label}: {score:.2f}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
62
63     return image, detected_objects
64
65 # Fungsi utama aplikasi
66 def main():
67     # Set judul aplikasi
68     st.title("Live Object Detection Qodrat")
69     # Tampilkan animasi di bawah sidebar
70     display_animation("Animation - 1734971782012.gif") # Path ke file GIF
71     st.sidebar.title("Menu")
72
73     # Atur background
74     set_background("background.png") # Path ke gambar latar belakang
75
76     model_path = "yolo11n.pt" # Path ke model YOLO Anda
77     model = load_model(model_path)
```

```

# Inisialisasi status deteksi
if "run_detection" not in st.session_state:
    st.session_state.run_detection = False

# Tombol mulai/berhenti deteksi
st.sidebar.markdown("### Kontrol Deteksi")
if st.sidebar.button("Mulai"):
    st.session_state.run_detection = True

if st.sidebar.button("Berhenti"):
    st.session_state.run_detection = False

# Jalankan deteksi saat aktif
if st.session_state.run_detection:
    cap = cv2.VideoCapture(0)
    st_frame = st.empty() # Placeholder untuk video
    st_detection_info = st.empty() # Placeholder untuk info deteksi

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

        # Proses deteksi dengan YOLO
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Konversi ke RGB
        results = model.predict(frame, imgsz=640)

        # Tampilkan hasil
        frame, detected_objects = display_results(frame, results[0])
        st_frame.image(frame, channels="RGB", use_column_width=True)

        # Tampilkan informasi deteksi
        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)

    cap.release()

if __name__ == "__main__":
    main()

```

Sebelum berlanjut jangan lupa untuk mengaktifkan venv kalian dengan cara seperti ini:

```

PS D:\Object Detection Project> cd .\venv
PS D:\Object Detection Project\venv> cd Scripts
PS D:\Object Detection Project\venv\Scripts> .\activate
(.venv) PS D:\Object Detection Project\venv\Scripts>

```

Kemudian kita install library di terminal kita :

```

(.venv) PS D:\Object Detection Project> pip install streamlit
(.venv) PS D:\Object Detection Project\venv> cd ..
(.venv) PS D:\Object Detection Project> pip install ultralytics

```

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

```

(.venv) PS D:\Object Detection Project> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.112.159:8501

2024-12-24 19:42:15.523 Examining the path of torch.classes raised: Tried to instantiate class '__path__._path',
ered 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 Menambahkan Latar Belakang

```
# Fungsi untuk menambahkan latar belakang
def set_background(image_path):
    with open(image_path, "rb") as file:
        base64_image = base64.b64encode(file.read()).decode()
    css = f"""
<style>
.stApp {{
    background-image: url("data:image;base64,{base64_image}");
    background-size: cover;
    background-repeat: no-repeat;
    background-attachment: fixed;
}}
</style>
"""
    st.markdown(css, unsafe_allow_html=True)
```

Tujuan: Mengatur gambar latar belakang pada aplikasi Streamlit.

Cara Kerja:

- 1) Membuka file gambar dalam mode **binary**.
- 2) Mengonversi gambar ke format **Base64** untuk menyematkannya di web.
- 3) Menggunakan **CSS** untuk mengatur gambar sebagai latar belakang aplikasi.

3. Fungsi untuk Menampilkan Animasi GIF di Sidebar

```
# Fungsi untuk menampilkan GIF di sidebar
def display_animation(gif_path):
    with open(gif_path, "rb") as file:
        gif_data = file.read()
    base64_gif = base64.b64encode(gif_data).decode()
    st.sidebar.markdown(
        f"""
        <div style="text-align: center;">
            
        </div>
        """,
        unsafe_allow_html=True,
    )
```

Tujuan: Menampilkan animasi GIF di sidebar aplikasi untuk meningkatkan estetika.

Cara Kerja:

- 1) Membaca file GIF dalam mode **binary**.
- 2) Mengonversi file GIF ke **Base64**.
- 3) Menampilkan GIF menggunakan elemen HTML ``.

4. 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

```
# Function untuk memproses hasil deteksi
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

```
# Fungsi utama aplikasi
def main():
    # Set judul aplikasi
    st.title("Live Object Detection Qodrat")
    # Tampilkan animasi di bawah sidebar
    display_animation("Animation - 1734971782012.gif") # Path ke file GIF
    st.sidebar.title("Live Object Detection Qodrat")

    # Atur background
    set_background("background.png") # Path ke gambar latar belakang

    model_path = "yolo11n.pt" # Path ke model YOLO Anda
    model = load_model(model_path)

    # Inisialisasi status deteksi
    if "run_detection" not in st.session_state:
        st.session_state.run_detection = False

    # Tombol mulai/berhenti deteksi
    st.sidebar.markdown("### MENU")
    if st.sidebar.button("Mulai"):
        st.session_state.run_detection = True

    if st.sidebar.button("Berhenti"):
        st.session_state.run_detection = False

    # Jalankan deteksi saat aktif
    if st.session_state.run_detection:
        cap = cv2.VideoCapture(0)
        st_frame = st.empty() # Placeholder untuk video
        st_detection_info = st.empty() # Placeholder untuk info deteksi

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

            # Proses deteksi dengan YOLO
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Konversi ke RGB
            results = model.predict(frame, imgsz=640)

            # Tampilkan hasil
            frame, detected_objects = display_results(frame, results[0])
            st_frame.image(frame, channels="RGB", use_column_width=True)

            # Tampilkan informasi deteksi
            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)

        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.

