

Nama : Faizal Kurniawan

NPM : 217064516061

Kelas : Pengolahan Citra (R. 03)

Berikut adalah langkah-langkah rinci untuk membuat dan menjalankan proyek **Deteksi Objek dengan YOLOv5 dan Streamlit**. Akan dibuat sebuah aplikasi web menggunakan **Streamlit**, yang akan memuat model **YOLOv5**, memproses gambar yang di-upload oleh pengguna, dan menampilkan hasil deteksi objek.

Langkah 1: Membuat Struktur Folder Proyek

1. Membuat Folder Proyek:

- Buat folder utama untuk proyek. Misalnya, beri nama folder tersebut **UAS FAIZAL**.

2. Struktur Folder Proyek:

- Di dalam folder **UAS FAIZAL**, buat struktur folder seperti berikut:

UAS FAIZAL/

— .env	# Virtual environment untuk proyek
— yolov5	# Folder untuk repository YOLOv5 (clone dari GitHub)
— app.py	# Script utama untuk aplikasi Streamlit
— requirements.txt	# Daftar library yang dibutuhkan
— yolov5s.pt	# Model YOLOv5 yang diunduh dan disimpan di sini

Langkah 2: Menginstall Aplikasi dan Extension yang Diperlukan

1. Menginstall VSCode dan Extensions:

- Install **Visual Studio Code (VSCode)**.
- Install extensions berikut di VSCode:
 - **Python**: untuk mendukung pengembangan Python.
 - **Streamlit**: untuk mempermudah menjalankan aplikasi Streamlit.

2. Install Python dan Virtual Environment:

- Pastikan **Python** sudah terpasang di sistem (untuk proyek ini, lebih baik menggunakan Python 3.7 atau yang lebih baru).
- Buat **virtual environment** untuk proyek ini dengan perintah:

```
"python -m venv .venv"
```

- Aktivasi environment:
 - **Windows:**

```
"\\.venv\\Scripts\\activate"
```
 - **Linux/Mac:**

```
"source .venv/bin/activate"
```

Langkah 3: Instalasi Dependencies (Daftar di requirements.txt)

1. **Buat file requirements.txt** di folder **UAS FAIZAL** dengan isi berikut:

```
torch  
streamlit  
pillow  
pandas
```

2. **Install dependencies:**

- Setelah virtual environment aktif, jalankan perintah berikut untuk menginstal dependencies yang terdaftar di requirements.txt:

```
"pip install -r requirements.txt"
```

Langkah 4: Clone Repository YOLOv5

1. **Clone YOLOv5 dari GitHub:**

- Untuk mendapatkan file YOLOv5 dan dependensinya, jalankan perintah berikut:

```
"git clone https://github.com/ultralytics/yolov5.git"
```
- Ini akan meng-clone repository YOLOv5 dari **Ultralytics** ke dalam folder **yolov5**.

2. **File yolov5s.pt:**

- Setelah menjalankan kode pertama kali, model **yolov5s.pt** akan diunduh secara otomatis ke dalam folder **UAS FAIZAL/**.

Langkah 5: Membuat dan Menulis Kode di app.py

1. **Buat file app.py** di dalam folder **UAS FAIZAL**.

2. **Menulis Kode di app.py:**

- Berikut adalah kode lengkap yang akan digunakan untuk aplikasi Streamlit.

```
import torch
import streamlit as st
import time
from PIL import Image
import numpy as np
import pandas as pd

# Load YOLOv5 pre-trained model langsung dari PyTorch Hub
model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)

st.title("Object Detection with YOLOv5")
st.write("Upload an image for object detection")

# Upload gambar
uploaded_file = st.file_uploader("Choose an image...", type=["jpg", "png", "jpeg"])

if uploaded_file is not None:
    start_time = time.time()

    # Membaca gambar dan mengonversinya ke format RGB menggunakan PIL
    img = Image.open(uploaded_file)
    img = img.convert("RGB") # Konversi ke format RGB

    # Mengonversi gambar PIL ke numpy array
    img = np.array(img)

    # Deteksi objek dengan YOLOv5
    results = model(img)
    elapsed_time = time.time() - start_time
```

```

# Tampilkan waktu inferensi
st.write(f'Inference Time: {elapsed_time:.4f} seconds')

# Render hasil deteksi objek
results.render()

st.image(results.ims[0], caption="Detected Objects",
use_container_width=True)

# Ambil hasil deteksi dalam format pandas DataFrame
detections = results.pandas().xywh[0]

# Menampilkan hasil deteksi dalam tabel
if len(detections) > 0:
    st.write(f'Detected {len(detections)} objects:')
    st.dataframe(detections[['xcenter', 'ycenter', 'width', 'height',
'confidence', 'name']])
else:
    st.warning("No objects detected in the image")

```

Penjelasan Kode app.py:

1. Mempersiapkan dan Memuat Model YOLOv5:

- Setelah mendapatkan file YOLOv5 dan dependensinya pada langkah 4, `torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)` digunakan untuk memuat model pre-trained YOLOv5 dari PyTorch Hub.

2. Antarmuka Pengguna dengan Streamlit:

- **st.file_uploader** memungkinkan pengguna untuk meng-upload gambar.
- **st.image** digunakan untuk menampilkan gambar hasil deteksi dengan bounding boxes yang digambar oleh YOLOv5.

3. Deteksi Objek dengan YOLOv5:

- Gambar yang di-upload akan diproses oleh model YOLOv5 untuk mendeteksi objek.
- Hasil deteksi objek (bounding boxes dan kelas objek) ditampilkan di tabel menggunakan **pandas**.

Langkah 6: Menjalankan Aplikasi

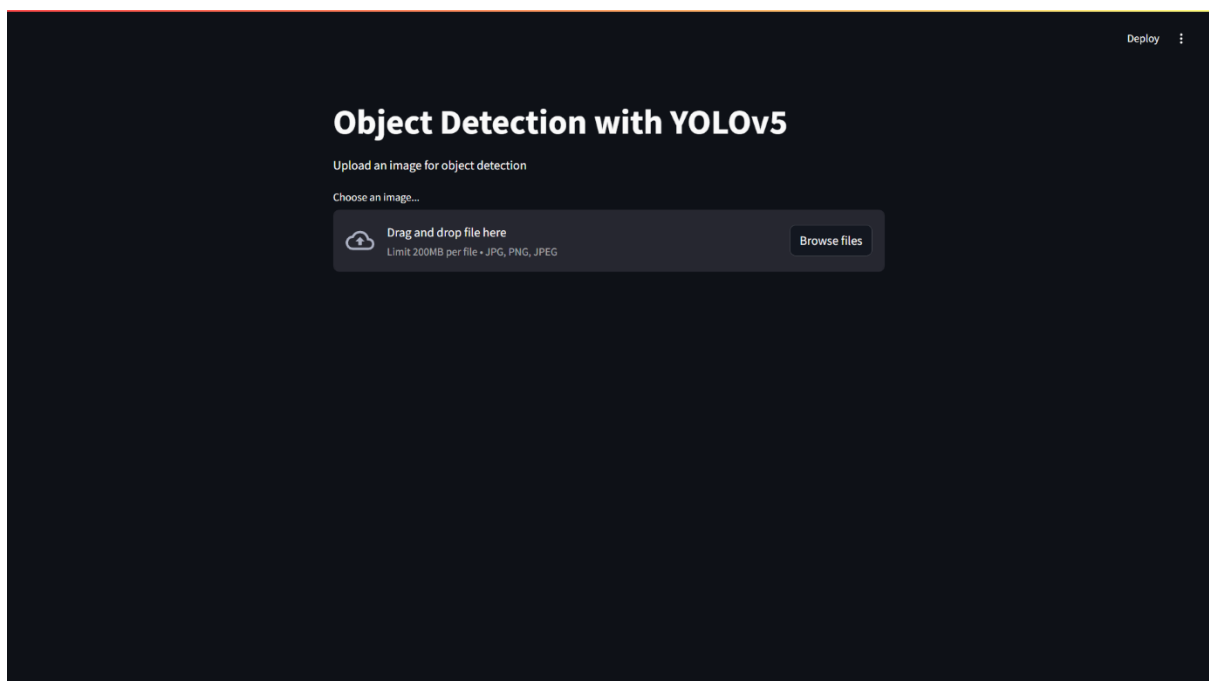
1. **Jalankan Aplikasi Streamlit:** Setelah menulis kode di **app.py**, jalankan aplikasi dengan perintah berikut di terminal:

“streamlit run app.py”

2. **Akses Aplikasi:**

- Buka browser dan akses aplikasi di alamat <http://localhost:8501/> untuk melihat aplikasi berjalan.
- Coba upload gambar yang berisi objek (misalnya orang, mobil, atau lainnya) dan lihat hasil deteksi objek.

Contoh tangkapan layar (screenshot) tampilan aplikasi web:



Object Detection with YOLOv5

Upload an image for object detection

Choose an image...

Drag and drop file here

Limit 200MB per file • JPG, PNG, GIF, JPEG

Browse files

images.jpg

63.0KB

X

Inference Time: 0.0889 seconds



Detected Objects

Detected 4 objects:

	xcenter	ycenter	width	height	confidence	name
0	390.9702	557.7148	107.8727	233.3172	0.8951	tie
1	897.2463	335.5977	593.7015	695.0706	0.8721	person
2	943.2167	544.7177	97.1711	258.8668	0.8713	tie
3	361.6765	544.9057	104.2082	647.6166	0.7965	person