

Nama : Serly Mulanty

Nim : 230741090

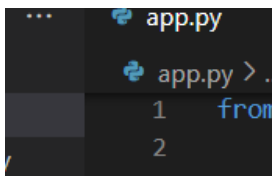
Prodi: Ilmu komputer

fakultas : Teknik dan sains

## 1. Buatlah folder

Buatlah folder baru untuk Proyek, Contoh : UAS KECERDASAN BUATAN

Masukkan file Yollo 11, lalu buatlah file.py untuk menjalankankode python.



## 2. Importing libraries

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

- YOLO: Digunakan untuk memuat model YOLO (You Only Look Once) yang digunakan untuk deteksi objek.
- cv2: Modul OpenCV untuk menangani gambar dan video.
- streamlit: Digunakan untuk membuat aplikasi web dengan antarmuka interaktif.
- PIL.Image: Modul untuk memproses gambar.
- numpy: Modul untuk bekerja dengan array multidimensi.
- collections.Counter: Digunakan untuk menghitung frekuensi kemunculan objek yang terdeteksi.

### 3. Menampilkan Hasil Deteksi

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

- Fungsi `load_model` digunakan untuk memuat model YOLO dari file `.pt` (file model terlatih).
- `@st.cache_resource` adalah decorator dari Streamlit yang memastikan bahwa model hanya dimuat sekali dan tidak diulang setiap kali aplikasi dijalankan. Ini meningkatkan efisiensi aplikasi.

### 4. Menampilkan Hasil Deteksi

```
# Process and display the detection results
def display_results(image, results):
    boxes = results.xyxy.cpu().numpy() # [x1, y1, x2, y2]
    scores = results.conf.cpu().numpy() # confidence scores
    labels = results.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}", (x2, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    return image, detected_objects
```

- Fungsi `display_results` digunakan untuk menampilkan hasil deteksi pada gambar (frame) dan menggambar kotak pembatas (bounding box) di sekitar objek yang terdeteksi.
- `results.xyxy` adalah koordinat dari kotak pembatas objek dalam format `[x1, y1, x2, y2]`.
- `results.conf` adalah skor kepercayaan (confidence score) untuk setiap deteksi objek.
- `results.cls` adalah indeks kelas objek yang terdeteksi.
- Hanya objek dengan skor kepercayaan lebih besar dari 0.5 yang akan digambar dan ditampilkan.
- `cv2.rectangle` menggambar kotak pembatas, dan `cv2.putText` menambahkan label dengan skor kepercayaan pada objek yang terdeteksi.

## 5. Menambahkan Latar Belakang kustom

```
# 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)
```

- Fungsi `set_background` digunakan untuk mengubah latar belakang aplikasi Streamlit dengan gambar yang dikodekan dalam format base64.
- Gambar pertama-tama dibaca sebagai byte dengan mode 'rb', kemudian dikodekan menjadi format base64.
- CSS disisipkan di dalam aplikasi Streamlit untuk mengatur gambar latar belakang (`background-image`) menggunakan data yang sudah dikodekan base64.

## 6. Fungsi Utama (Main Function)

```
# Main Streamlit app
def main():
    set_background("image.jpg")
    st.title("Serly Mulanty")
    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

            # Show 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:
```

- Fungsi utama `main` adalah pusat dari aplikasi Streamlit.
- `set_background("image.jpg")` menambahkan gambar latar belakang ke aplikasi.
- `st.title("Serly Mulanty")` menampilkan judul aplikasi di bagian atas.
- `st.sidebar.title("Settings")` menambahkan judul di sidebar aplikasi untuk pengaturan.
- Model YOLO dimuat dengan `load_model("yolo11n.pt")`.

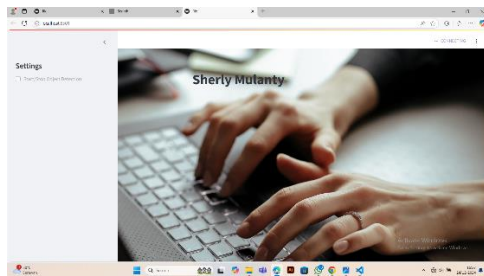
- `st.sidebar.checkbox("Start/Stop Object Detection")` menambahkan checkbox untuk mengaktifkan atau menonaktifkan deteksi objek.
- Jika checkbox aktif (`run_detection`), aplikasi akan membuka kamera (menggunakan `cv2.VideoCapture(0)`) untuk menangkap video.
- Di dalam loop, frame video diambil dan diproses menggunakan YOLO untuk deteksi objek.
- Hasil deteksi digambar pada frame, dan objek yang terdeteksi akan dihitung frekuensinya menggunakan Counter.
- Hasil deteksi ditampilkan di aplikasi Streamlit dalam bentuk gambar dan teks.

## 7. Menjalankan Aplikasi

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

- Ini adalah bagian yang memastikan bahwa fungsi `main()` dijalankan jika file ini dieksekusi langsung. Dengan kata lain, aplikasi Streamlit akan dimulai di sini.

## 8. Jika telah berhasil lanjut saja Deteksi real-time!



Objek yang terdeteksi akan ditampilkan seperti contoh dibawah ini.

