

Nama : Nadia Sasti

NIM : 230741095

Matkul : Kecerdasan Buatan

Penjelasan uas

#Langkah Pertama

.venv:

Folder ini biasanya digunakan untuk menyimpan virtual environment Python. Virtual environment adalah lingkungan terisolasi untuk mengelola dependensi proyek Python.

Coco.names

COCO names merujuk pada nama-nama kelas (kategori objek) yang digunakan dalam dataset COCO (Common Objects in Context). Dataset COCO adalah salah satu dataset besar yang sering digunakan dalam pelatihan model deteksi objek, segmentasi, dan captioning gambar.

deteksi_wajah.py:

File Python yang berisi skrip utama untuk menjalankan deteksi wajah. Di dalamnya, mungkin terdapat kode yang menggunakan pustaka seperti OpenCV atau YOLO untuk mendeteksi wajah.

haarcascade_eye.xml:

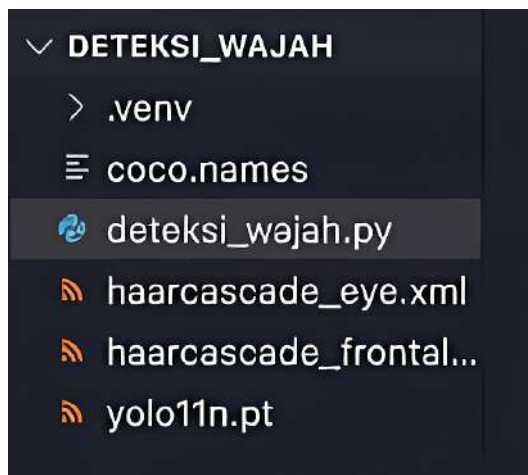
Ini adalah file XML dari model Haar Cascade yang digunakan untuk mendeteksi mata. Model ini biasanya digunakan dengan pustaka OpenCV.

haarcascade_frontal... (kemungkinan besar haarcascade_frontalface_default.xml):

File XML lain yang digunakan untuk mendeteksi wajah manusia. Ini juga bagian dari model Haar Cascade dan digunakan dalam algoritme deteksi wajah dengan OpenCV.

yolo11.pt:

File model berbasis YOLO (You Only Look Once), biasanya digunakan untuk deteksi objek yang lebih canggih dibandingkan Haar Cascade. File ini berisi bobot model (weights) yang digunakan untuk mendeteksi wajah atau objek lainnya.



#Langkah Kedua

import torch
Modul ini mengimpor pustaka PyTorch, yang merupakan framework machine learning open-source. PyTorch sering digunakan untuk deep learning dan berbagai tugas AI.

import cv2
Modul ini mengimpor pustaka OpenCV, yang digunakan untuk pemrosesan gambar dan video. OpenCV mendukung berbagai operasi seperti deteksi wajah, pemrosesan gambar, dan manipulasi video.

import numpy as np
Numpy adalah pustaka Python untuk komputasi numerik. Dengan as np, kita dapat menggunakan np sebagai alias pendek untuk memanggil fungsi-fungsi Numpy. Contohnya, np.array untuk membuat array.

import streamlit as st
Streamlit adalah framework untuk membuat aplikasi web interaktif dengan Python. Dengan as st, pustaka ini diberi alias pendek st untuk digunakan dalam kode.

from PIL import Image
Modul ini mengimpor kelas Image dari pustaka Pillow (PIL). Pillow adalah pustaka untuk memanipulasi dan memproses gambar.

import tempfile
Modul tempfile digunakan untuk membuat file atau direktori sementara yang akan dihapus otomatis setelah tidak digunakan.

import os
Modul os digunakan untuk operasi sistem seperti mengelola file dan direktori. Misalnya, mengakses path, membuat direktori baru, atau membaca variabel lingkungan.

```
import torch
import cv2
import numpy as np
import streamlit as st
from PIL import Image
import tempfile
import os
```

#Langkah ketiga

model = torch.load('yolov1n.pt')

- Fungsi: Baris ini menggunakan fungsi torch.load dari pustaka PyTorch untuk memuat model YOLOv1 yang sudah dilatih sebelumnya.
- Parameter: 'yolov1n.pt' adalah nama file model (file dengan ekstensi .pt menunjukkan model PyTorch). Pastikan file ini berada di direktori yang sesuai atau gunakan path lengkap jika file berada di lokasi lain.
- Catatan: Anda bisa mengganti 'yolov1n.pt' dengan nama file model lain jika menggunakan model yang berbeda.

model.eval()

- Fungsi: Perintah ini menetapkan model ke "mode evaluasi".
- Tujuan: Mode evaluasi memastikan bahwa layer-layer seperti dropout atau batch normalization tidak aktif (karena layer-layer ini hanya diperlukan selama pelatihan, bukan inferensi).
- Catatan: Mode evaluasi penting untuk mendapatkan hasil prediksi yang konsisten saat menggunakan model.

Komentar

- Komentar dalam kode menjelaskan setiap langkah:
 - "Muatan model YOLOv1" menunjukkan model yang dimuat adalah YOLOv1.
 - "Ganti dengan path file model .pt Anda" berarti Anda perlu memastikan path file model benar.
 - "Set model ke mode evaluasi" memberi tahu tujuan penggunaan model.eval().

```
# Muat model YOLOv11 (pastikan file model YOLOv11 sudah ada di direktori yang benar)
model = torch.load('yolo11n.pt') # Ganti dengan path file model .pt Anda
model.eval() # Set model ke mode evaluasi
```

#Langkah keempat

Muat	Classifier	Haar	Cascade
Digunakan untuk mendeteksi wajah menggunakan model Haar Cascade bawaan OpenCV. Model ini dilatih untuk mengenali wajah frontal pada gambar atau video.			

Inisialisasi

Streamlit

Menambahkan antarmuka pengguna dengan Streamlit, seperti judul aplikasi dan deskripsi. Ini membantu pengguna memahami fungsi aplikasi, yaitu deteksi wajah dan objek dari gambar atau video.

Fungsi detect_objects

- Deteksi Wajah: Frame diubah menjadi gambar grayscale, lalu fungsi detectMultiScale digunakan untuk mendeteksi wajah berdasarkan classifier Haar Cascade.
- Deteksi Objek dengan YOLOv1:
 - Frame dikonversi menjadi tensor PyTorch dengan format yang sesuai.
 - Piksel gambar dinormalisasi ke rentang 0-1.
 - Tensor diproses oleh model YOLOv1 untuk mendeteksi objek, tanpa menghitung gradien (hemat memori).

Fungsi ini menggabungkan deteksi wajah dan deteksi objek berbasis YOLOv1 untuk pengolahan gambar atau video.

```
# Muat classifier Haar Cascade untuk deteksi wajah
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

# Inisialisasi Streamlit
st.title("Deteksi Wajah dan Objek (Handphone) dari Foto, Video, dan Kamera Langsung")
st.write("Unggah foto/video atau pilih kamera untuk mendeteksi objek seperti handphone dan wajah.")

# Fungsi untuk deteksi objek dan wajah menggunakan YOLOv11
def detect_objects(frame):
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

    # Deteksi objek menggunakan YOLOv11
    img_tensor = torch.from_numpy(frame).permute(2, 0, 1).float() # Convert frame to tensor
    img_tensor /= 255.0 # Normalisasi gambar
    img_tensor = img_tensor.unsqueeze(0) # Menambah dimensi batch (1, C, H, W)

    with torch.no_grad():
        outputs = model(img_tensor) # Deteksi objek
```

#Langkah kelima

Proses Hasil Deteksi:

outputs[0]: Mengambil hasil deteksi dari model.

pred_classes: Mengambil kelas dari hasil deteksi (biasanya ID kelas dari objek yang terdeteksi), kemudian diubah menjadi tipe integer.

pred_boxes: Mengambil koordinat kotak pembatas (bounding box) dari objek terdeteksi.

Pengelolaan Bounding Box

frame.shape: Mendapatkan dimensi gambar/video, yaitu tinggi, lebar, dan jumlah saluran (biasanya 3 untuk RGB).

Variabel kosong seperti **boxes**, **class_ids**, dan **confidences** disiapkan untuk menyimpan informasi tentang bounding box, ID kelas, dan tingkat kepercayaan deteksi.

Iterasi untuk Memproses Setiap Deteksi

- Melakukan iterasi pada setiap deteksi.
- Memfilter deteksi berdasarkan ID kelas tertentu (misalnya 0 untuk objek tertentu, seperti handphone).
- Mengubah koordinat bounding box relatif (0–1) ke dimensi piksel sesuai dengan ukuran gambar.
- Menyimpan bounding box, tingkat kepercayaan, dan ID kelas ke dalam daftar masing-masing.

Non-Maximum Suppression (NMS)

- NMS digunakan untuk mengurangi duplikasi deteksi pada area yang sama.
- Parameter:
 - **boxes**: Bounding box dari deteksi.
 - **confidences**: Tingkat kepercayaan deteksi.
 - **0.5**: Threshold untuk confidence score.
 - **0.4**: Threshold untuk overlapping (Intersection over Union/IoU).

Menampilkan Deteksi

Menampilkan bounding box:

- Mengambil koordinat dan dimensi dari bounding box.
- Menambahkan teks label kelas dan confidence score.
- Menggambar kotak pembatas dan teks pada frame menggunakan `cv2.rectangle` dan `cv2.putText`

```
# Proses hasil deteksi
results = outputs[0]
pred_classes = results[:, :-1].cpu().numpy().astype(int)
pred_boxes = results[:, :-1].cpu().numpy()

height, width, _ = frame.shape
boxes = []
class_ids = []
confidences = []

for i in range(len(pred_classes)):
    if pred_classes[i] == 0: # Misalnya, '0' untuk handphone (sesuaikan dengan ID kelas yang diinginkan)
        x1, y1, x2, y2 = pred_boxes[i]
        x1, y1, x2, y2 = int(x1 * width), int(y1 * height), int(x2 * width), int(y2 * height)
        boxes.append([x1, y1, x2 - x1, y2 - y1])
        confidences.append(results[i][4].cpu().item())
        class_ids.append(pred_classes[i])

# NMS untuk mengurangi deteksi berlebih
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

# Gambar kotak dan label pada objek yang terdeteksi
if len(indexes) > 0:
    for i in indexes.flatten():
        x, y, w, h = boxes[i]
        label = str(pred_classes[i])
        confidence_text = f'{label} {(confidences[i]*100:.2f)}%'
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(frame, confidence_text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0, 0), 2)
```

#Langkah keenam

Deteksi Wajah

Kode ini mendeteksi wajah pada frame menggunakan bounding box:

- **Iterasi pada wajah yang terdeteksi**: Koordinat bounding box diperoleh untuk setiap wajah.
- **Menggambar bounding box**: Setiap wajah diberi kotak pembatas (rectangle) menggunakan warna merah.
- **Menambahkan label**: Teks "Wajah Terdeteksi" ditampilkan di atas bounding box. Outputnya adalah frame yang telah ditambahkan bounding box dan label.

Deteksi Objek pada Foto

Fungsi ini bertugas untuk mendeteksi objek dari gambar yang diunggah:

- Memuat gambar: File gambar diubah menjadi array menggunakan library seperti PIL dan NumPy.
- Proses deteksi objek: Fungsi pemrosesan mendeteksi objek dari gambar yang sudah diubah ke array.
- Mengembalikan hasil deteksi: Gambar dengan bounding box dan label objek dikembalikan.

Pemrosesan Video

Fungsi ini memproses file video yang diunggah untuk mendeteksi objek:

- **Membuat file sementara:** Video yang diunggah disimpan sementara di disk.
- **Membuka video:** Video dibaca menggunakan OpenCV.
- **Iterasi frame per frame:**
 - Setiap frame dari video dibaca.
 - Jika video selesai atau frame tidak terbaca, iterasi dihentikan.
 - Objek dan wajah pada setiap frame dideteksi menggunakan fungsi yang telah didefinisikan.
- **Output:** Frame dengan deteksi objek dan wajah ditampilkan atau disimpan.

```
# Deteksi wajah
for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
    cv2.putText(frame, "Wajah Terdeteksi!", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0, 0), 2)

return frame

# Fungsi untuk deteksi pada foto
def detect_from_image(uploaded_image):
    image = Image.open(uploaded_image)
    frame = np.array(image)
    detected_frame = detect_objects(frame)
    return detected_frame

# Fungsi untuk streaming video dari file
def process_video(video_file):
    # Membuat file sementara untuk video yang diunggah
    with tempfile.NamedTemporaryFile(delete=False) as temp_file:
        temp_file.write(video_file.read())
        temp_file.close() # Simpan file sementara

    # Membaca video dari file sementara
    cap = cv2.VideoCapture(temp_file.name)

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        # Deteksi objek dan wajah pada frame
        detected_frame = detect_objects(frame)
```

#Langkah ketujuh

Membaca Video dan Menampilkan Deteksi pada Streamlit

Kode ini memproses video frame per frame dan menampilkan hasil deteksinya menggunakan Streamlit:

- Membuka file video sementara: Video dibuka dengan cv2.VideoCapture.
- Iterasi frame per frame:
 - Frame dibaca dari video.

- o Jika frame tidak tersedia atau video selesai, proses dihentikan.
 - o Deteksi objek dilakukan pada frame menggunakan fungsi `detect_objects`.
- Konversi format warna:
 - o OpenCV menggunakan format warna BGR secara default, yang perlu diubah ke RGB agar kompatibel dengan Streamlit.
 - o Gambar diubah menjadi format Image agar dapat ditampilkan.
- Menampilkan hasil di Streamlit:
 - o Streamlit digunakan untuk menampilkan frame hasil deteksi dengan fungsi `st.image`.
 - o Opsi untuk menghentikan video diberikan melalui tombol "Stop Video".
- Membersihkan sumber daya: Setelah proses selesai, video dilepas menggunakan `cap.release`.

Fungsi Deteksi dari Kamera Langsung

Fungsi ini memproses input video dari kamera secara langsung:

- **Membuka kamera:**
 - o Kamera dibuka dengan `cv2.VideoCapture(0)`, di mana 0 menunjukkan perangkat kamera utama.
- **Iterasi secara terus-menerus:**
 - o Frame dari kamera dibaca secara real-time.
 - o Jika frame tidak terbaca, proses dihentikan.
 - o Deteksi objek dilakukan untuk setiap frame secara langsung.
- **Hentikan proses jika diperlukan:** Proses akan terus berjalan hingga kondisi tertentu (biasanya dengan tombol "Stop" atau penghentian manual).

```
# Membaca video dari file sementara
cap = cv2.VideoCapture(temp_file.name)

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Deteksi objek dan wajah pada frame
    detected_frame = detect_objects(frame)

    # Convert frame ke format Image agar bisa ditampilkan oleh Streamlit
    img = Image.fromarray(cv2.cvtColor(detected_frame, cv2.COLOR_BGR2RGB))

    # Tampilkan gambar pada Streamlit
    st.image(img, channels="RGB", use_column_width=True)

    # Tunggu sejenak agar tampilan bisa di-update
    if st.button("Stop Video"):
        break

cap.release()

# Fungsi untuk deteksi kamera langsung
def detect_from_camera():
    cap = cv2.VideoCapture(0) # Buka kamera
    while True:
        ret, frame = cap.read()
        if not ret:
            break
```

#Langkah kedelapan

Deteksi Objek dan Wajah pada Frame

- **Fungsi detect_objects(frame):** Deteksi objek dan wajah dilakukan pada setiap frame video atau input dari kamera.
- **Konversi warna:** Hasil deteksi (frame) yang menggunakan format warna BGR (OpenCV) dikonversi ke RGB agar kompatibel dengan Streamlit.
- **Menampilkan hasil:** Hasil deteksi ditampilkan dalam Streamlit menggunakan fungsi st.image.

Tombol "Stop Kamera"

- **Penghentian streaming kamera:** Jika tombol "Stop Kamera" diklik, maka proses membaca frame dari kamera akan dihentikan dengan perintah break.
- **Membersihkan sumber daya:** Kamera dilepaskan menggunakan cap.release() untuk memastikan tidak ada konflik saat kamera digunakan kembali.

Pilihan Input untuk Deteksi

Bagian ini menyediakan antarmuka untuk memilih input berupa foto, video, atau kamera langsung:

- **Unggahan Foto:**
 - File foto dapat diunggah dalam format .jpg, .jpeg, atau .png.
 - Fungsi detect_from_image(uploaded_image) akan memproses foto untuk mendeteksi objek atau wajah.
 - Hasil deteksi ditampilkan di Streamlit dengan label "Hasil Deteksi Foto".
- **Unggahan Video:**
 - File video dapat diunggah dalam format .mp4, .avi, atau .mov.
 - Fungsi process_video(uploaded_video) akan memproses video dan menampilkan hasil deteksi frame per frame.
- **Kamera Langsung:**
 - Jika tidak ada file yang diunggah, pengguna dapat memilih untuk menggunakan kamera langsung dengan menekan tombol "Deteksi Kamera Langsung".
 - Fungsi detect_from_camera() akan membaca video dari kamera secara real-time dan menampilkan hasil deteksi.

Penanganan Kondisi Input

- Jika tidak ada file yang diunggah atau tombol kamera langsung tidak ditekan, Streamlit akan menampilkan pesan: "Unggah foto atau video, atau pilih untuk menggunakan kamera langsung."

Fungsi main()

- Fungsi ini mengatur seluruh alur aplikasi:
 - Menangani unggahan file atau pilihan kamera langsung.
 - Memanggil fungsi terkait untuk memproses dan menampilkan hasil deteksi.
- Penutup program diatur dengan if __name__ == "__main__": main() untuk memastikan fungsi main() hanya dijalankan ketika file ini dijalankan sebagai program utama.


```

# Deteksi objek dan wajah pada frame
detected_frame = detect_objects(frame)

# Convert frame ke format Image agar bisa ditampilkan oleh Streamlit
img = Image.fromarray(cv2.cvtColor(detected_frame, cv2.COLOR_BGR2RGB))

# Tampilkan gambar pada Streamlit
st.image(img, channels="RGB", use_column_width=True)

# Tunggu sejenak agar tampilan bisa di-update
if st.button("Stop Kamera"):
    break

cap.release()

# Bagian untuk memilih input (Foto, Video, atau Kamera Langsung)
uploaded_image = st.file_uploader("Pilih foto untuk diunggah", type=["jpg", "jpeg", "png"])
uploaded_video = st.file_uploader("Pilih video untuk diunggah", type=["mp4", "avi", "mov"])

# Menjalankan deteksi sesuai input yang dipilih
if uploaded_image is not None:
    detected_image = detect_from_image(uploaded_image)
    st.image(detected_image, caption="Hasil Deteksi Foto", channels="RGB", use_column_width=True)

elif uploaded_video is not None:
    process_video(uploaded_video)

else:
    if st.button("Deteksi Kamera Langsung"):
        detect_from_camera()
    else:
        st.write("Unggah foto atau video, atau pilih untuk menggunakan kamera langsung.")

if __name__ == "__main__":
    main()

```

#Langkah kesembilan

Setelah semua selesai kita ke terminal dan menjalankan deploy

```

(hnms_venv) nadiasuci@MacBook-Air:~$ streamlit run deteksi_wajah.py

You can now view your Streamlit app in your browser:

Link URL: https://33aea17a0b18961
Network URL: http://172.20.10.4:8501

For better performance, install the WebRTC module:
$ sudo apt-get install --install
$ sudo systemctl restart

Confidence: 100.00%
2024-12-21 21:57:08.455 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 100.00%
2024-12-21 21:57:47.369 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 100.00%
2024-12-21 21:57:49.884 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 100.00%
2024-12-21 21:57:50.338 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:50.492 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 100.00%
2024-12-21 21:57:52.210 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:52.265 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:52.319 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:52.373 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:52.427 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:52.481 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:54.349 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 100.00%
2024-12-21 21:57:54.373 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:55.289 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:55.893 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:56.461 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:56.478 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:56.478 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%
2024-12-21 21:57:56.477 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Please utilize the 'use_container_width' parameter instead.
Confidence: 99.99%

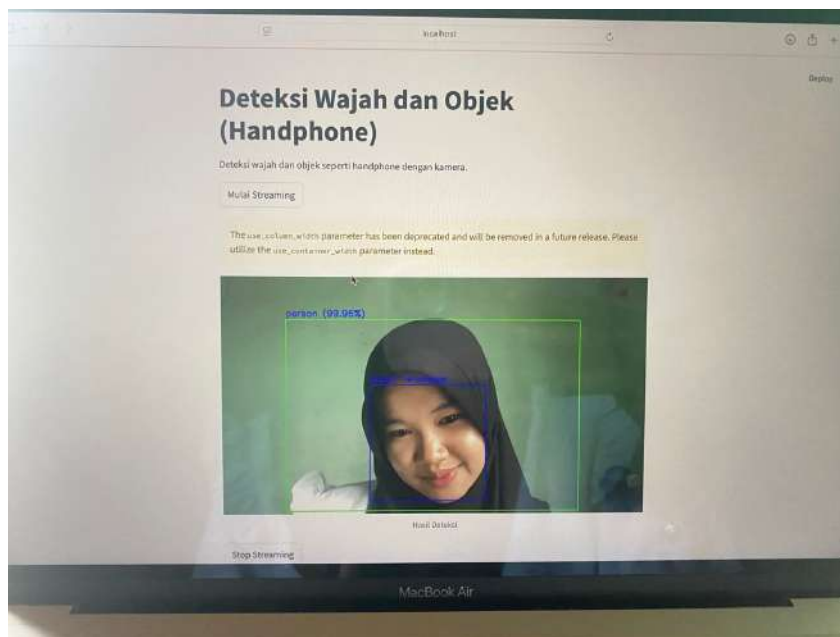
DE RUN | DEPLOY STREAMLIT

```

(DEPLOY kamera langsung)



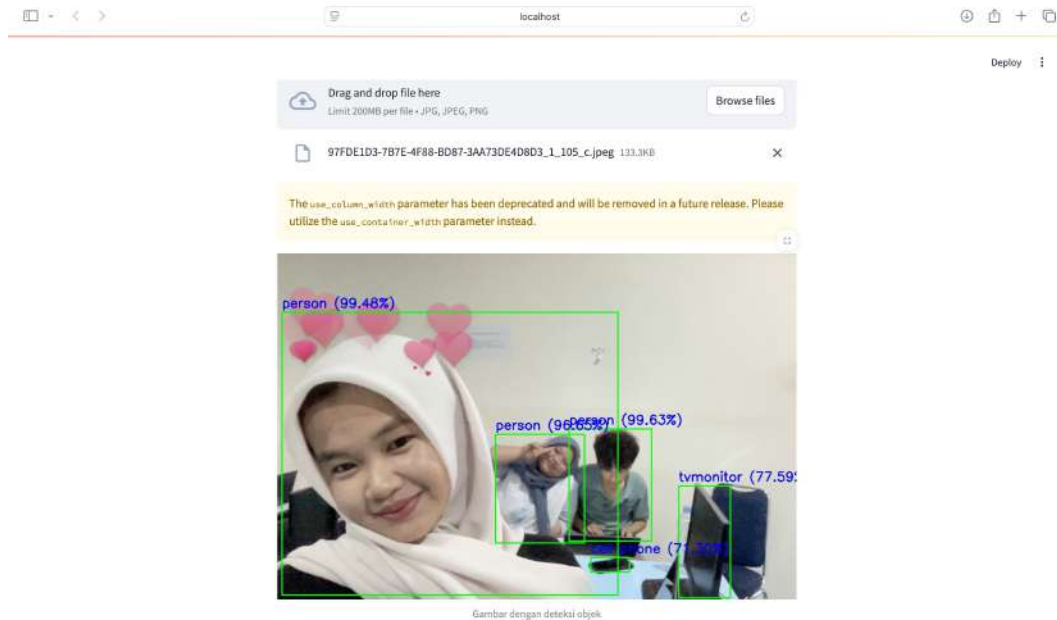
(Hasil deploy kamera langsung)



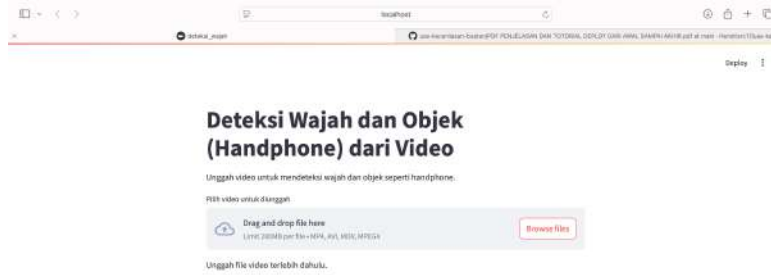
(DEPLOY dengan objek gambar)



(hasil DEPLOY deteksi objek gambar)



(DEPLOY dengan objek video)



(hasil DEPLOY dengan objek video)

