

Tugas UTS Komputasi Intelegensia

Nailiyatul Mammunah
NPM: 2106722801

November 2, 2024

1 Pendahuluan

Laporan ini bertujuan untuk menjelaskan penerapan deteksi objek menggunakan model YOLOv5 di Google Colab. Video yang digunakan dalam eksperimen ini berjudul `bottle-detection.mp4` dan berisi adegan dengan beberapa botol. Video ini dipilih untuk menunjukkan kemampuan YOLOv5 dalam mendeteksi objek dalam lingkungan terkontrol. Proses yang dibahas meliputi pengunggahan file video, pemuatan model YOLOv5, deteksi objek pada setiap frame, pra-pemrosesan hasil deteksi, pelacakan objek antar frame, serta optimasi performa YOLOv5 untuk deteksi real-time.

2 Tinjauan Pustaka

YOLO (You Only Look Once) adalah metode deteksi objek berbasis jaringan konvolusional yang dikembangkan untuk mendeteksi objek secara cepat dan akurat. Berbeda dengan model lain seperti Faster R-CNN yang menggunakan jaringan proposal region, YOLO langsung memprediksi bounding box dan label kelas dalam satu tahap komputasi. Hal ini membuat YOLO sangat cepat dan ideal untuk aplikasi real-time. YOLOv5 adalah versi terbaru yang dikembangkan oleh Ultralytics, dengan varian seperti YOLOv5s yang ringan dan cepat, cocok untuk pemrosesan video secara langsung di platform berbasis cloud seperti Google Colab.

3 Metodologi

Bagian ini menjelaskan tahapan yang dilakukan untuk mengunggah video, memuat model YOLOv5, deteksi objek pada setiap frame, pra-pemrosesan data, pelacakan objek, dan optimasi performa model.

3.1 Pengunggahan Video dan Akses Google Drive di Colab

Di Google Colab, terdapat dua metode utama untuk memproses video dengan YOLOv5: pengunggahan langsung dari perangkat lokal atau akses Google Drive.

Pengunggahan Langsung Video dapat diunggah langsung dari perangkat pengguna ke Colab dengan kode berikut:

```
from google.colab import files
uploaded = files.upload()
```

Akses Google Drive File video juga dapat disimpan di Google Drive dan diakses dengan cara berikut:

```
from google.colab import drive
drive.mount('/content/drive')
```

Setelah terhubung, file dapat diakses melalui path Google Drive, misalnya `/content/drive/MyDrive/bottle-detection.mp4`.

3.2 Memuat Model YOLOv5

Model YOLOv5 diunduh dan dimuat menggunakan perintah berikut:

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
!pip install -qr requirements.txt
import torch
model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)
```

Kode ini mendownload repositori YOLOv5 dan memuat model pra-latih `yolov5s` yang ringan.

3.3 Cara YOLO Mendeteksi Objek dan Menghasilkan Output Video

YOLO bekerja dengan membagi setiap frame dari video menjadi grid dan melakukan prediksi bounding box serta label kelas pada setiap sel grid. Model ini melakukan deteksi secara langsung dengan menggunakan jaringan konvolusional, yang memungkinkan YOLO mendeteksi beberapa objek dalam satu tahap komputasi tanpa perlu jaringan proposal region.

Setiap frame diproses secara independen, di mana YOLO akan menghasilkan bounding box dan label kelas untuk objek yang dikenali dalam frame tersebut. Output ini kemudian digunakan untuk menggambar kotak pembatas (bounding box) di sekitar objek dan menambahkan label kelas yang terdeteksi di atasnya.

Setiap frame hasil deteksi disimpan sebagai frame baru, yang kemudian disusun kembali menjadi video output.

Berikut adalah langkah-langkah teknis yang digunakan untuk menghasilkan output video:

1. **Proses Frame demi Frame:** Video dipecah menjadi frame individu, dan setiap frame diproses oleh YOLOv5 untuk mendeteksi objek.
2. **Penggambaran Bounding Box dan Label:** Pada setiap frame, bounding box ditambahkan pada koordinat objek yang terdeteksi, serta label kelas dan skor kepercayaan.
3. **Penyusunan Ulang Frame Menjadi Video:** Setelah semua frame diproses, mereka digabungkan kembali menjadi video output, yang memperlihatkan deteksi objek secara real-time.

3.4 Pra-pemrosesan Hasil Deteksi

Setelah objek terdeteksi, hasilnya perlu dinormalisasi dengan mengubah koordinat bounding box menjadi skala $[0, 1]$ berdasarkan ukuran frame video. Hasil deteksi kemudian disimpan dalam format terstruktur untuk memudahkan pelacakan di seluruh frame.

3.5 Pelacakan Objek

Pelacakan objek dilakukan untuk menjaga konsistensi identitas objek yang sama di beberapa frame. Ini memungkinkan objek untuk diikuti secara akurat sepanjang pergerakannya dalam video. Metode pelacakan seperti *Kalman Filter* atau *Sort-Based Tracking* dapat digunakan untuk melacak objek secara real-time.

3.6 Optimasi Performa YOLOv5

Beberapa teknik optimasi diterapkan untuk meningkatkan performa YOLOv5 di Colab:

- **Resolusi Input:** Mengurangi resolusi input untuk mempercepat proses inferensi.
- **Frame Skipping:** Mengabaikan beberapa frame untuk mengurangi beban komputasi.
- **Kuantisasi Model:** Menggunakan kuantisasi untuk mengurangi ukuran model dan waktu inferensi.

4 Hasil dan Pembahasan

Model YOLOv5 berhasil mendeteksi objek botol pada setiap frame video `bottle-detection.mp4`. Setiap objek yang terdeteksi ditampilkan dengan bounding box dan label kelas yang menunjukkan jenis objek serta skor kepercayaan. Hal ini memungkinkan visualisasi yang jelas dan informatif dalam video output.

Dalam video output, setiap frame menunjukkan hasil deteksi YOLOv5, dengan kotak pembatas yang menyorot setiap botol yang terdeteksi. Teknik ini sangat efektif dalam aplikasi real-time, karena memungkinkan pengguna untuk melihat objek yang terdeteksi secara langsung pada rekaman video. Dengan memanfaatkan optimasi seperti pengurangan resolusi input dan frame skipping, pemrosesan video dapat dipercepat tanpa mengorbankan akurasi.

5 Kesimpulan

Penerapan YOLOv5 di Google Colab memungkinkan deteksi objek secara real-time dengan akurasi yang tinggi. Metode ini sangat berguna untuk aplikasi yang membutuhkan respons cepat, seperti sistem pengawasan atau analisis video. Dalam eksperimen ini, YOLOv5 menunjukkan kinerja yang andal dalam mendeteksi botol pada video `bottle-detection.mp4`, dan optimasi tambahan mampu meningkatkan kecepatan pemrosesan secara signifikan.

6 Daftar Pustaka

- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv preprint, arXiv:1804.02767. Retrieved from <https://arxiv.org/abs/1804.02767>
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv preprint, arXiv:2004.10934. Retrieved from <https://arxiv.org/abs/2004.10934>
- Ultralytics. (2021). *YOLOv5*. GitHub repository. Retrieved from <https://github.com/ultralytics/yolov5>