



KELOMPOK 1

Deteksi objek kaca mata menggunakan metode YOLO (You Only Look Once)

MACHINE LEARNING

TEKNIK INFORMATIKA
22A



ANGGOTA

AKMAL NUUR FAUZAN

20220040060

ABDUL RAHMAN JAINUN

20220040157

DHIKA DWI NUGRAHA

20220040285

GIDEON CHANDRA DWIANTO

20220040153

KANIA PURNARHAYU

20220040099

MUHAMMAD RAIHAN ASSHAFWAT

20220040137

RIVAN MOCHAMMAD FIRZAN

20220040132



PENGERTIAN

Ultralytics YOLOv8 adalah versi terbaru dari model deteksi objek dan segmentasi gambar YOLO (You Only Look Once) yang dikembangkan oleh Ultralytics. Model YOLOv8 dirancang agar cepat, akurat, dan mudah digunakan, menjadikannya pilihan yang sangat baik untuk berbagai tugas deteksi objek dan segmentasi gambar. Ini dapat dilatih pada kumpulan data besar dan mampu berjalan pada berbagai platform perangkat keras, dari CPU hingga GPU. Algoritma YOLO merupakan algoritma deep learning untuk mendeteksi objek yang menggunakan pendekatan berbeda dari algoritma lain, yaitu menerapkan sebuah jaringan syaraf tunggal pada keseluruhan citra



PENGERTIAN

Algoritma YOLO menggunakan convolutional neural network (CNN) untuk mendeteksi objek secara real-time. Seperti namanya, algoritma ini hanya membutuhkan satu propagasi maju melalui jaringan saraf untuk mendeteksi objek.

Ini berarti bahwa prediksi di seluruh gambar dilakukan dalam satu kali algoritma. CNN digunakan untuk memprediksi berbagai probabilitas kelas dan kotak pembatas secara bersamaan.



Mengapa algoritma YOLO itu penting

Algoritma YOLO penting karena alasan berikut:

- Kecepatan: Algoritma ini meningkatkan kecepatan deteksi karena dapat memprediksi objek secara real-time.
- Akurasi tinggi: YOLO adalah teknik prediktif yang memberikan hasil akurat dengan kesalahan latar belakang minimal.
- Kemampuan belajar: Algoritme memiliki kemampuan pembelajaran yang sangat baik yang memungkinkannya mempelajari representasi objek dan menerapkannya dalam deteksi objek.



Bagaimana algoritma YOLO bekerja

Algoritma YOLO bekerja menggunakan tiga teknik berikut:

- Blok sisa adalah gambar dibagi menjadi beberapa grid. Setiap grid memiliki dimensi $S \times S$.
- Regresi kotak pembatas adalah garis besar yang menyoroti objek dalam gambar. Setiap kotak pembatas pada gambar terdiri dari atribut berikut:
 1. Lebar (bb)
 2. Tinggi (bh)
 3. Kelas (misalnya, orang, mobil, lampu lalu lintas, dll.)–
Dilambangkan dengan huruf c.
 4. Pusat kotak pembatas (bx,by)
- Persimpangan Selama Union (IOU) adalah fenomena dalam deteksi objek yang menjelaskan bagaimana kotak saling tumpang tindih.

TAHAP PERTAMA

```
1 + from ultralytics import YOLO
2 + import cv2
3 + import time
4 +
```

-
1. 'from ultralytics import YOLO': Ini mengimpor kelas YOLO dari pustaka Ultralytics. YOLO adalah model deteksi objek yang cepat dan efisien.
 2. 'import cv2': Ini mengimpor OpenCV, sebuah pustaka populer untuk komputer visi dan pemrosesan gambar.
 3. 'import time': Ini mengimpor modul waktu, yang akan digunakan untuk mengukur waktu yang diperlukan untuk melakukan deteksi objek.
-


```
4 +  
5 + model = YOLO("best.pt")  
6 +  
7 + cam = cv2.VideoCapture(0)  
8 + if not cam.isOpened():  
9 +     raise("No Camera")
```

-
1. 'model = YOLO("best.pt")': Ini memuat model YOLO dari file "best.pt". "best.pt" adalah berkas yang berisi parameter-parameter yang telah dilatih sebelumnya dari model YOLO. Model ini akan digunakan untuk mendeteksi objek dalam gambar atau video.
 2. 'cam = cv2.VideoCapture(0)': Ini membuka kamera dengan nomor identifikasi 0 (biasanya kamera utama). Fungsi cv2.VideoCapture() digunakan untuk mengakses aliran video dari perangkat kamera. Hasilnya adalah objek VideoCapture yang dapat digunakan untuk mengambil frame dari aliran video.
 3. 'if not cam.isOpened()': Ini melakukan pemeriksaan apakah kamera telah berhasil dibuka. Jika kamera tidak terbuka (mungkin karena tidak ada kamera terhubung), maka akan muncul pengecualian yang mengatakan "No Camera". Hal ini digunakan untuk memastikan bahwa program tidak mencoba melakukan deteksi objek tanpa akses ke kamera.
-


```
10 +  
11 + while True:  
12 +     ret, image = cam.read()  
13 +     if not ret:  
14 +         break  
15 +
```

Kode ini menambahkan loop `while True:` untuk terus-menerus membaca frame dari kamera dan melakukan deteksi objek pada setiap frame:

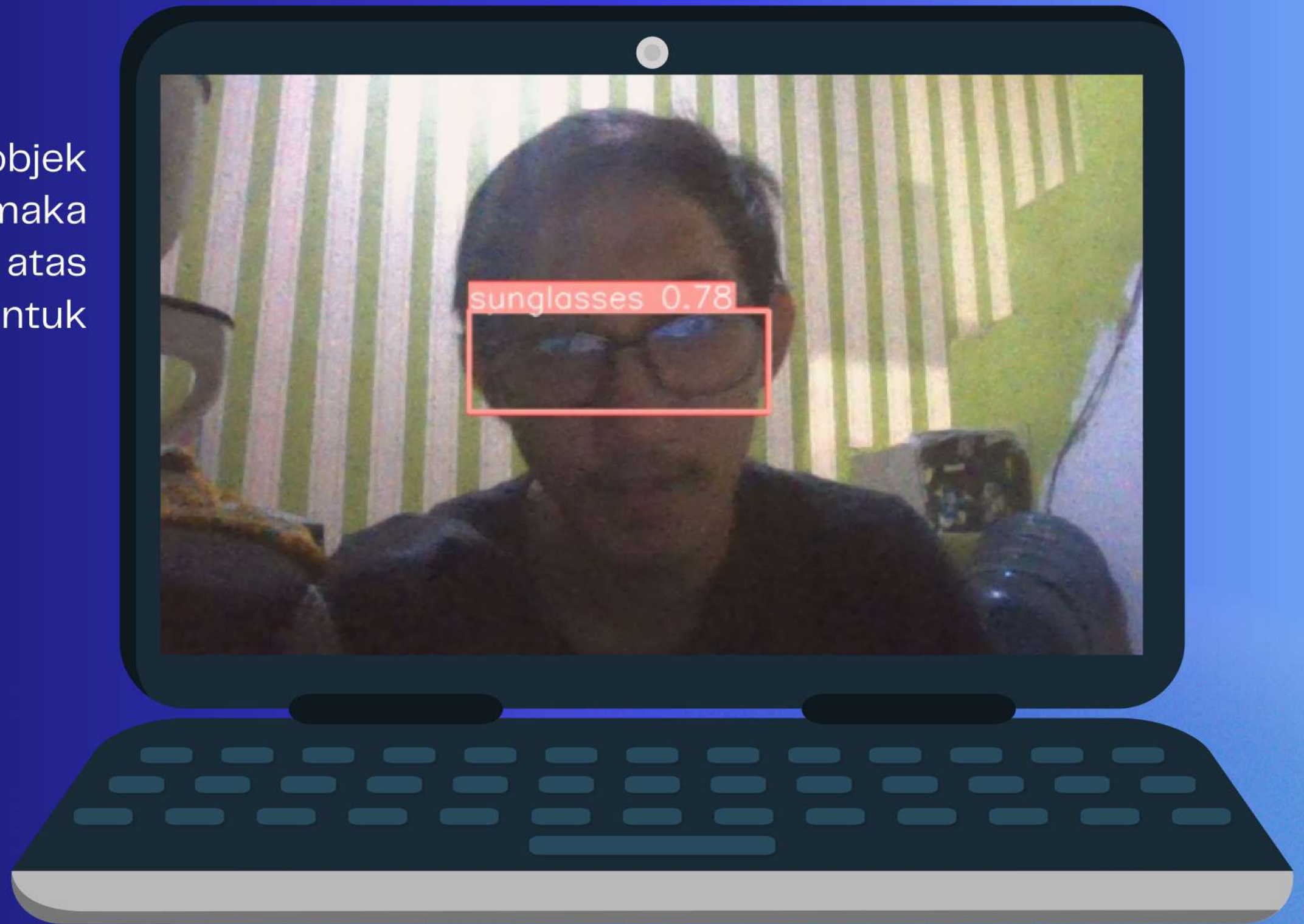
1. `'ret, image = cam.read()'` : Ini membaca satu frame dari aliran video yang diberikan oleh objek `cam`. Fungsi `read()` mengembalikan dua nilai, yaitu nilai boolean `ret` yang menunjukkan apakah pembacaan frame berhasil (`True` jika berhasil, `False` jika tidak), dan `image` yang merupakan frame yang dibaca dari kamera.
 2. `'if not ret: break'` : Ini memeriksa apakah pembacaan frame berhasil. Jika tidak berhasil (nilai `ret` adalah `False`), loop akan dihentikan dengan `break`. Hal ini menandakan bahwa aliran video telah berakhir atau terjadi kesalahan saat membaca frame, sehingga tidak perlu dilakukan deteksi objek lagi.
-

Kode ini menambahkan beberapa langkah lagi setelah membaca frame dari kamera dan sebelum menghentikan program :

1. `'_time_mulai = time.time()'` : Ini mengambil waktu mulai sebelum melakukan prediksi objek pada frame.
 2. `'result = model.predict(image, show=True)'` : Ini menggunakan model YOLO yang telah dimuat sebelumnya untuk memprediksi objek dalam gambar. image adalah frame yang telah dibaca dari kamera. show=True digunakan untuk menampilkan hasil deteksi objek langsung di atas frame. Hasilnya disimpan dalam variabel result.
 3. `'print("waktu", time.time() - _time_mulai)'` : Ini mencetak waktu yang dibutuhkan untuk melakukan prediksi objek pada satu frame dengan mengurangi waktu mulai dari waktu selesai.
 4. `'_key = cv2.waitKey(1)'` : Ini menunggu input dari pengguna selama 1 milidetik. Jika tombol "q" ditekan, loop akan dihentikan.
 5. `'cam.release()'` : Ini melepaskan aliran video dari kamera. Ini penting dilakukan setelah selesai menggunakan kamera untuk memastikan sumber daya sistem dibebaskan.
 6. `'cv2.destroyAllWindows()'` : Ini menutup semua jendela OpenCV yang terbuka. Ini dilakukan setelah selesai menggunakan aplikasi.
-

OUTPUT

- Jika model YOLO berhasil mendeteksi objek pada frame yang dibaca dari kamera, maka hasil deteksi objek akan ditampilkan di atas frame, seperti kotak pembatas dan label untuk objek yang terdeteksi.





TERIMA KASIH