



Enhancing Object Detection with YOLO and OpenCV in Python

Kelompok 2 - Cucur Adabi
Kelas Kecerdasan Buatan (F)

Kelompok Cucur Adabi



Rizky Alifiyah Rahma

5025211208



Salsabila Fatma Aripa

5025211057



Tsabita Putri Ramadhany

5025211130

DAFTAR ISI

01

BACKGROUND

Latar belakang Object Detection with YOLO and OpenCV in Python

02

OBJECT DETECTION INTRODUCTION

Desain dan fitur yang digunakan pada Object Detection

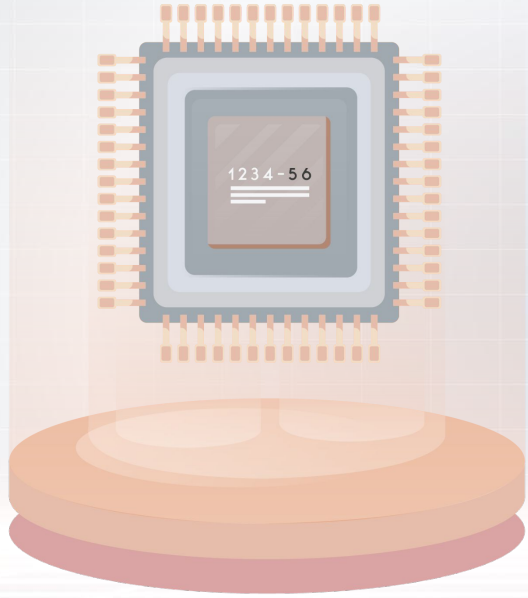
03

DEVELOPMENT

Algoritma dan Langkah Implementasi

ENHANCING OBJECT DETECTION!

“Menaklukkan Dunia dengan Ketepatan dan Kecepatan!”



01 BACKGROUND

Artificial Intelligence / Kecerdasan Buatan

Artificial Intelligence

AI adalah bidang yang berkembang pesat dalam dunia komputer dan teknologi. AI memungkinkan komputer dan sistem untuk melakukan tugas-tugas yang sebelumnya hanya dapat dilakukan oleh manusia, seperti pengambilan keputusan, pemecahan masalah, dan pembelajaran. Dalam konteks "Object Detection", AI dapat memberikan kontribusi penting dalam menciptakan program secara akurat dan efisien mendeteksi objek.



ARTIFICIAL INTELLIGENCE

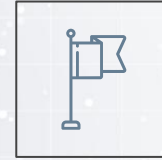
AI telah membawa perubahan yang signifikan, terutama dalam bidang deteksi objek. Deteksi objek merupakan salah satu aplikasi penting dalam AI yang memungkinkan komputer untuk mengenali dan memahami objek dalam gambar atau video. Dengan menggunakan teknik yang canggih seperti YOLO (You Only Look Once) dan OpenCV, deteksi objek dapat dilakukan dengan presisi tinggi dan kecepatan yang mengagumkan. Hal ini memungkinkan sistem komputer untuk secara otomatis mengidentifikasi berbagai objek, seperti manusia, mobil, atau hewan, dan memberikan informasi yang berguna dalam berbagai bidang, mulai dari keamanan, transportasi, hingga pengolahan gambar dan video. Dengan kemampuan ini, AI semakin memperluas batasan kemampuan komputer dan memberikan solusi yang efektif dan efisien dalam memahami dunia visual di sekitar kita.

OBJECT DETECTION



YOLO (You Only Look Once)

Pendekatan deteksi objek real-time yang cepat dan efisien. Melakukan deteksi objek dengan melihat seluruh gambar sekaligus, memberikan kotak pembatas dan label objek dengan tingkat kepastian. Tingkat kepastian ini menunjukkan sejauh mana sistem yakin bahwa objek tersebut benar-benar ada dalam gambar.



OpenCV (Open Source Computer Vision)

Digunakan untuk memanipulasi dan menganalisis gambar, serta menggambar kotak pembatas dan label pada objek yang terdeteksi oleh YOLO. OpenCV memberikan kemampuan untuk memproses dan memanipulasi gambar dengan berbagai teknik seperti pembuatan blob, perhitungan koordinat, dan penggambaran grafis.

Menghasilkan sistem deteksi objek yang lebih canggih, efisien, dan dapat diimplementasikan dengan mudah dalam proyek-proyek pengolahan gambar dan computer vision.

– OBJECT DETECTION GOAL

MEKANISME "Enhancing Object Detection with YOLO and OpenCV in Python"

1. Memuat model dan konfigurasi YOLO dengan menggunakan library OpenCV.
2. Mendefinisikan fungsi untuk mendeteksi objek dalam gambar.
3. Membuat blob dari gambar yang akan dideteksi.
4. Melakukan deteksi objek menggunakan YOLO dengan memasukkan blob ke dalam model.
5. Mengolah hasil deteksi untuk mendapatkan informasi objek yang terdeteksi.
6. Melakukan non-maximum suppression untuk menghilangkan deteksi yang tumpang tindih.

MEKANISME "Enhancing Object Detection with YOLO and OpenCV in Python"

7. Menggambar kotak pembatas dan label objek pada gambar hasil deteksi.
8. Menghitung tinggi dan lebar objek dan menyimpan informasi objek dalam bentuk dictionary.
9. Memulai video capture dan membaca setiap frame dari video.
10. Mendeteksi objek dalam setiap frame menggunakan fungsi deteksi objek.
11. Menampilkan frame hasil deteksi dengan kotak pembatas dan label objek.
12. Menampilkan informasi objek seperti nama, tinggi, dan lebar.

02

OBJECT DETECTION INTRODUCTION

Desain dan fitur yang
digunakan pada Object
Detection



Object Detection Design

Berfokus pada peningkatan deteksi objek menggunakan algoritma YOLO (You Only Look Once) dan library OpenCV di Python. Kode ini menggunakan model YOLO yang telah dilatih sebelumnya dan file konfigurasi (yolov3.weights dan yolov3.cfg) untuk mendeteksi objek pada gambar atau frame video. Desainnya melibatkan proses memuat model dan konfigurasi, mendefinisikan fungsi untuk deteksi objek, memproses hasil deteksi, dan menampilkan frame yang telah diberi anotasi dengan objek yang terdeteksi.

Object Detection Features

FITUR	PENJELASAN
Memuat Model	Memuat model YOLO dan file konfigurasinya menggunakan fungsi <code>cv2.dnn.readNet</code> . File berisi bobot (weights) YOLO (<code>yolov3.weights</code>) yang berisi parameter-parameter yang telah dipelajari, sementara file konfigurasi (<code>yolov3.cfg</code>) mendefinisikan arsitektur jaringan.
Deteksi Objek	Memproses gambar dengan membuat blob, mengatur blob sebagai input untuk jaringan, dan mendapatkan hasil deteksi. Fungsi ini juga mengambil ID kelas, kepercayaan (confidence), dan koordinat kotak pembatas untuk objek yang terdeteksi.

Object Detection Features

FITUR	PENJELASAN
Non-Maximum Suppression	Untuk menghilangkan deteksi yang tumpang tindih, menggunakan fungsi <code>cv2.dnn.NMSBoxes</code> . Fungsi ini menyaring kotak pembatas yang tumpang tindih berdasarkan kepercayaan dan luas tumpang tindih.
Anotasi dan Ekstraksi Informasi	Menggambar kotak pembatas dan label di sekitar objek yang terdeteksi. Kode ini menggunakan label kelas dari file "coco.names" untuk memberi label yang sesuai pada objek. Selain itu, kode ini juga menghitung tinggi dan lebar setiap objek dan menyimpan informasi tersebut dalam sebuah daftar.

Object Detection Features

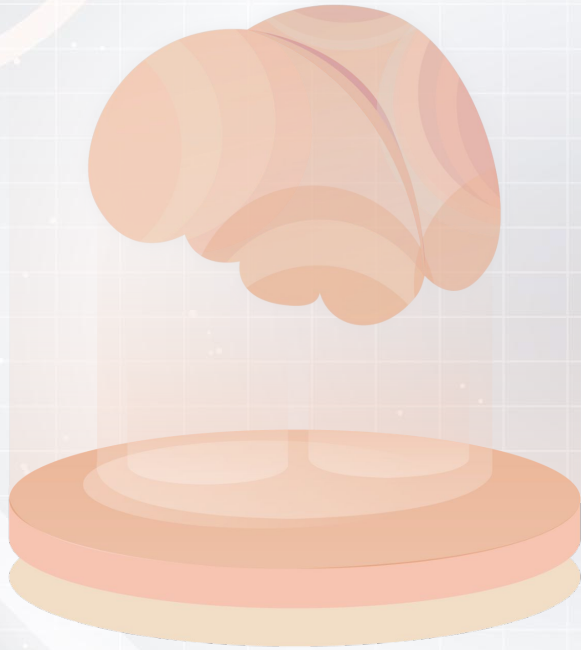
FITUR	PENJELASAN
Pemrosesan dan Tampilan Video	<p>Menggunakan fungsi VideoCapture dari OpenCV untuk membaca frame dari file video ("in.avi"). Program ini terus-menerus membaca frame, melakukan deteksi objek pada setiap frame, dan menampilkan frame yang telah diberi anotasi dalam jendela dengan nama "Deteksi Objek". Kode ini juga mencetak informasi (label kelas, tinggi, dan lebar) dari objek yang terdeteksi.</p>



03

DEVELOPMENT

Algorithm and Implementation Step



ALGORITHM

Di dalam **Enhancing Object Detection with YOLO and OpenCV in Python** akan dilakukan deteksi object. Lalu, bagaimana langkah-langkah umum untuk meningkatkan deteksi objek menggunakan YOLO dan OpenCV ?

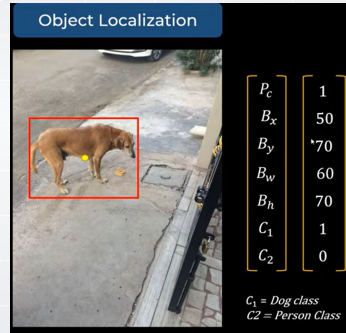
YOLO ALGORITHM

Algoritma YOLO (You Only Look Once) merupakan algoritma deteksi objek yang populer dalam bidang pengolahan citra dan komputer visi. Algoritma YOLO menggunakan pendekatan yang berbeda dari metode deteksi objek tradisional. **Dalam deteksi objek tradisional**, algoritma akan memindai gambar secara bertahap dengan menggunakan algoritma seperti **sliding window**. Namun, **YOLO** mengubah masalah deteksi objek menjadi masalah **regresi** di mana objek dan **kotak pembatasnya** diprediksi secara **simultan dalam satu langkah**.

YOLO membagi gambar menjadi grid dan setiap grid memprediksi beberapa **bounding box (kotak pembatas)** dan kelas objek yang terkait. Kemudian, menggunakan regresi logistik, YOLO memperbaiki koordinat dan ukuran bounding box untuk mendapatkan deteksi yang lebih akurat. Selain itu, YOLO menggunakan **non-maximum suppression** untuk menghilangkan deteksi ganda dan meningkatkan akurasi.

Object Localization

Object localization, dalam konteks pengolahan citra dan komputer visi, merujuk pada tugas **menemukan lokasi** atau **posisi objek** di dalam sebuah gambar.



P_c	Apakah <i>video</i> atau <i>image</i> merupakan objek yang ingin dideteksi?
B_x	Koordinat titik pusat sumbu X objek
B_y	Koordinat titik pusat sumbu Y objek
B_w	Lebar <i>bounding box</i> objek
B_h	Tinggi <i>bounding box</i> objek
C_1	Jenis objek yang ingin dideteksi (anjing)
C_2	Jenis objek yang ingin dideteksi (manusia)

YOLO ALGORITHM AND OPENCV



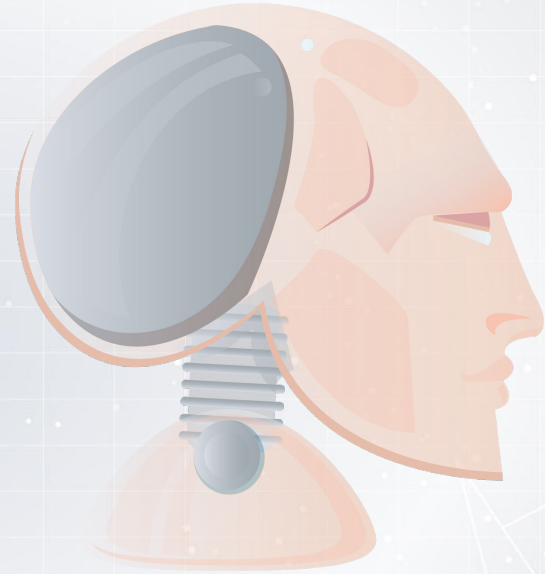
Keuntungan utama dari YOLO adalah **kecepatannya**. YOLO dapat melakukan deteksi objek secara **real-time** pada video atau streaming dengan **kecepatan yang tinggi**. Algoritma YOLO juga **cukup akurat** dalam mendeteksi berbagai jenis objek dalam berbagai **konteks visual**. Dalam implementasinya menggunakan **OpenCV sebagai Library**.

Algoritma YOLO menggunakan **OpenCV** sebagai library untuk membaca, memproses, dan menampilkan gambar atau video. OpenCV menyediakan fitur yang memungkinkan integrasi dengan algoritma YOLO, seperti membaca dan menampilkan gambar/video, pemrosesan gambar sebelum dan setelah deteksi objek, serta visualisasi hasil deteksi dengan menampilkan bounding box dan label objek.



IMPLEMENTATION STEP

Secara garis besar, Langkah - langkah dalam implementasi Final Project kali ini adalah sebagai berikut ;



IMPLEMENTATION STEP

Step	Keterangan
Import Library	Mengimpor modul OpenCV (cv2) dan modul NumPy (numpy) yang diperlukan untuk pemrosesan citra dan manipulasi array.
Load model dan konfigurasi YOLO <code>net = cv2.dnn.readNet</code>	Membaca file model YOLO (yolov3.weights) dan file konfigurasi (yolov3.cfg). Selain itu, kode juga membaca file coco.names yang berisi daftar kelas objek yang dapat dideteksi oleh model YOLO.

File "**yolov3.weights**" berisi parameter-parameter yang mencakup bobot (weights) yang digunakan untuk menghitung dan memprediksi probabilitas keberadaan objek pada setiap grid di gambar.

file "**yolov3.cfg**" digunakan untuk memuat konfigurasi model YOLOv3 untuk menginisialisasi arsitektur jaringan dalam model dan menentukan bagaimana model akan beroperasi selama proses deteksi objek.

IMPLEMENTATION STEP

Step	Keterangan
Layer Output <code>layer_names = net.getLayerNames()</code>	Mendapatkan informasi tentang nama-nama layer dalam model YOLO serta nama-nama layer yang merupakan output dari model tersebut.
Fungsi <code>detect_objects(image)</code>	Mendeteksi objek dalam gambar. Nantinya akan mengambil gambar/video sebagai input dan mengembalikan gambar/video dengan kotak pembatas dan label objek yang terdeteksi, serta informasi objek seperti tinggi dan lebar .

IMPLEMENTATION STEP

Step	Fungsi
Video Capture <code>cap = cv2.VideoCapture('video file')</code>	Membuka file video untuk diproses frame demi frame.
Looping	Membaca frame dari video menggunakan cap.read() . Jika frame berhasil dibaca, fungsi detect_objects() dipanggil untuk mendeteksi objek dalam frame. Frame hasil deteksi ditampilkan menggunakan cv2.imshow() , dan informasi objek dicetak pada konsol.
Membersihkan dan Menutup Jendela	Setelah keluar dari loop, video capture dihentikan dengan cap.release() . Semua jendela yang dibuka ditutup dengan cv2.destroyAllWindows() .



THANK YOU