



Implementasi dan Pengembangan Sistem Pengenalan Plat Nomor Kendaraan Secara Otomatis Menggunakan Yolo v5 dan *Google Vision* OCR

Sunarta¹, M Iqbal Erlangga Prabowo Adi Putra², M. David Chaidir³, Rizqullah Aryaputra Piliang³,
Andrea Songklanaita⁴

^{1,2,3,4}Universitas Pertahanan Republik Indonesia

E-mail: narta.laut@gmail.com, miqbalerlangga123@gmail.com davidchaidir53@gmail.com,
risqullaharya16@gmail.com, akuandreasongklanaita@gmail.com

Article Info	Abstract
Article History Received: 2025-05-13 Revised: 2025-06-23 Published: 2025-07-06 Keywords: ALPR; YOLOv5; Google Vision; OCR; License Plate Detection; Character Recognition.	Automatic License Plate Recognition (ALPR) is an important technology in various applications, such as traffic surveillance, access control, and automated payment systems. This study develops an ALPR system that combines the YOLOv5 object detection model and Google Vision-based Optical Character Recognition (OCR) technology. The dataset used consists of 5,564 Indonesian vehicle license plate images, which have gone through an augmentation process to increase data diversity and model resilience to variations in lighting, viewing angles, and weather conditions. YOLOv5 is used to detect the position of the license plate, while Google Vision OCR is used to recognize the license plate characters more accurately. Valid license plate detection results are stored in a MySQL database to support efficient data management. Based on testing on test data that has never been seen by the model, this system shows a high level of accuracy in license plate detection and recognition, with a processing speed that meets the needs of real-time applications. The proposed system is able to overcome detection challenges in dynamic and complex environmental conditions, and provides better results than conventional methods.
Artikel Info Sejarah Artikel Diterima: 2025-05-13 Direvisi: 2025-06-23 Dipublikasi: 2025-07-06 Kata kunci: ALPR; YOLOv5; Google Vision; OCR; Deteksi Plat Nomor; Pengenalan Karakter.	Abstrak Pengenalan Plat Nomor Otomatis (Automatic License Plate Recognition/ALPR) merupakan teknologi penting dalam berbagai aplikasi, seperti pengawasan lalu lintas, kontrol akses, dan sistem pembayaran otomatis. Penelitian ini mengembangkan sistem ALPR yang menggabungkan model deteksi objek YOLOv5 dan teknologi Optical Character Recognition (OCR) berbasis Google Vision. Dataset yang digunakan terdiri dari 5.564 citra plat nomor kendaraan Indonesia, yang telah melalui proses augmentasi untuk meningkatkan keragaman data dan ketahanan model terhadap variasi pencahayaan, sudut pandang, dan kondisi cuaca. YOLOv5 digunakan untuk mendeteksi posisi plat nomor, sedangkan Google Vision OCR digunakan untuk mengenali karakter plat nomor secara lebih akurat. Hasil deteksi plat nomor yang valid disimpan dalam database MySQL guna mendukung efisiensi pengelolaan data. Berdasarkan pengujian pada data uji yang belum pernah dilihat oleh model, sistem ini menunjukkan tingkat akurasi tinggi dalam deteksi dan pengenalan plat nomor, dengan kecepatan pemrosesan yang memenuhi kebutuhan aplikasi real-time. Sistem yang diusulkan mampu mengatasi tantangan deteksi dalam kondisi lingkungan yang dinamis dan kompleks, serta memberikan hasil yang lebih baik dibandingkan metode konvensional.

I. PENDAHULUAN

Pengenalan Plat Nomor Otomatis (*Automatic License Plate Recognition/ALPR*) telah menjadi salah satu teknologi yang esensial dalam berbagai aplikasi, seperti pengawasan lalu lintas, manajemen parkir, sistem tol elektronik, dan kontrol akses di area publik. ALPR memungkinkan sistem untuk mendeteksi dan mengenali plat nomor kendaraan secara otomatis melalui kamera, yang dapat diintegrasikan dengan sistem manajemen lalu lintas atau keamanan. Namun, tantangan yang muncul dalam pengembangan sistem ALPR mencakup akurasi deteksi dan kecepatan

pengenalan karakter, terutama dalam kondisi lingkungan yang menantang seperti pencahayaan yang buruk, sudut pengambilan gambar yang beragam, serta adanya gangguan visual seperti kotoran atau refleksi.

Dalam beberapa tahun terakhir, metode deteksi objek berbasis deep learning telah menunjukkan performa yang signifikan dalam berbagai aplikasi computer vision. Salah satu metode yang paling menonjol adalah *You Only Look Once* (YOLO), sebuah model convolutional neural network (CNN) yang dikenal karena kemampuannya melakukan deteksi objek secara real-time dengan akurasi tinggi. YOLO v5, versi

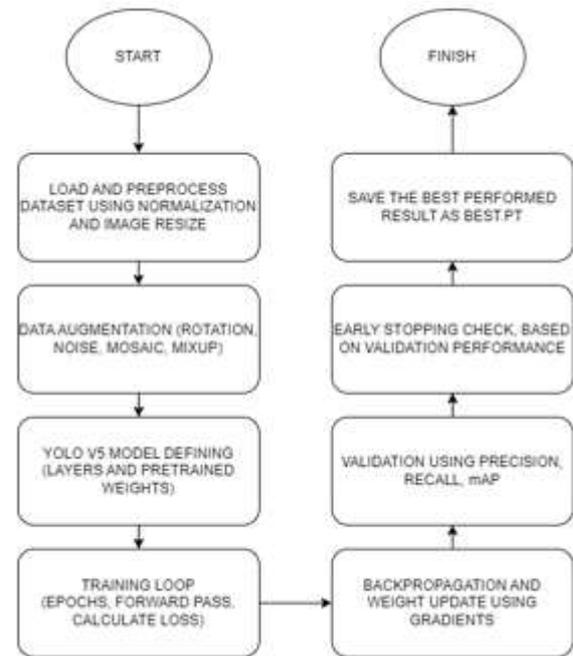
terbaru dari metode ini, menawarkan beberapa keunggulan dibandingkan versi sebelumnya, seperti YOLOv3 dan YOLOv4. Keunggulan-keunggulan ini mencakup kecepatan pemrosesan yang lebih tinggi, ukuran model yang lebih ringan, serta kemudahan implementasi dengan framework modern seperti PyTorch. YOLO v5 juga dilengkapi dengan teknik augmentasi data yang lebih canggih, yang memungkinkan model untuk lebih adaptif terhadap variasi lingkungan dan kondisi pencahayaan, menjadikannya pilihan ideal untuk diaplikasikan pada sistem ALPR yang beroperasi di dunia nyata.

Selain deteksi plat nomor, proses *Optical Character Recognition* (OCR) memainkan peran krusial dalam mengenali karakter plat nomor yang terdeteksi. Di antara berbagai teknologi OCR yang tersedia, *Google Vision* OCR terbukti unggul dalam hal akurasi pengenalan teks pada berbagai jenis dokumen dan gambar, termasuk dalam kondisi kualitas gambar yang bervariasi. *Google Vision* OCR memiliki kemampuan untuk mengenali karakter dengan baik meskipun citra plat nomor terpengaruh oleh variasi pencahayaan atau sudut pengambilan gambar. Oleh karena itu, kombinasi YOLO v5 untuk deteksi plat nomor dan *Google Vision* OCR untuk pengenalan karakter diharapkan dapat menghasilkan sistem ALPR yang lebih akurat dan efisien dalam berbagai kondisi lingkungan.

Penelitian ini bertujuan untuk mengembangkan sistem ALPR berbasis YOLO v5 dan *Google Vision* OCR yang mampu bekerja secara real-time dengan akurasi tinggi. Dalam penelitian ini, akan dibahas penerapan kedua teknologi tersebut, serta evaluasi performa sistem yang diusulkan berdasarkan pengujian di lingkungan nyata. Sistem ini diharapkan dapat meningkatkan efisiensi pengenalan plat nomor kendaraan, khususnya dalam aplikasi di bidang transportasi dan keamanan, sekaligus mengatasi tantangan yang muncul dalam kondisi lingkungan yang dinamis dan kompleks.

II. METODE PENELITIAN

Penelitian ini mengembangkan sistem *Automatic License Plate Recognition* (ALPR) yang menggunakan YOLOv5 untuk deteksi plat nomor dan *Google Vision* OCR untuk pengenalan karakter. Metode ini mencakup beberapa tahap penting, yaitu akuisisi data, preprocessing dan augmentasi data, pelatihan model, validasi, dan implementasi sistem ALPR. Setiap tahap didukung dengan elemen teknis dan visualisasi melalui flowchart untuk membantu pemahaman proses.



Gambar 1. Model Creation FLOWCHART

Dataset yang digunakan dalam sistem ini terdiri dari 5.564 citra plat nomor kendaraan Indonesia. Data ini dibagi menjadi tiga subset: 5.102 gambar untuk pelatihan, 432 untuk validasi, dan 21 untuk pengujian. Pengumpulan data dilakukan dengan memperhatikan variasi format plat nomor di Indonesia, sehingga model dapat belajar dan mengenali karakteristik visual plat nomor secara umum. Proporsi besar pada data pelatihan dimaksudkan untuk meningkatkan kemampuan generalisasi model, sedangkan data validasi dan pengujian digunakan untuk memantau dan mengevaluasi performa model.



Gambar 2. Pembagian Dataset dalam Proses Akuisisi Data

Proses augmentasi data bertujuan untuk memperluas keragaman dataset, yang melibatkan berbagai teknik seperti rotasi, translasi, perubahan skala, serta modifikasi hue, saturasi, kecerahan, dan shear. Selain itu, teknik mosaic dan mixup digunakan untuk menggabungkan beberapa citra secara acak dalam satu batch, memperkaya variasi yang dihadirkan pada model. Augmentasi ini memastikan bahwa model tetap dapat mengenali plat nomor dalam berbagai kondisi visual, termasuk variasi pencahayaan, cuaca, dan sudut pandang.

Model YOLOv5 digunakan untuk deteksi plat nomor. Arsitektur YOLOv5 disesuaikan melalui konfigurasi file YAML, yang mendefinisikan lapisan dan parameter yang diperlukan untuk model. Proses ini memanfaatkan bobot pretrained dari dataset umum seperti COCO untuk mempercepat pelatihan melalui transfer learning, di mana model memanfaatkan pengetahuan awal dari dataset lain untuk mendeteksi pola dasar dalam citra plat nomor. Konfigurasi file menunjukkan jalur direktori untuk dataset pelatihan, validasi, dan pengujian, serta menetapkan kategori deteksi dengan nama 'License_Plate'.

```
train: C:/Users/risqu/Desktop/PLATE/yolov5/dataset/train/images
val: C:/Users/risqu/Desktop/PLATE/yolov5/dataset/valid/images
test: C:/Users/risqu/Desktop/PLATE/yolov5/dataset/test/images
nc: 0
names: ['License_Plate']
```

Gambar 3. Konfigurasi Dataset untuk Pelatihan Model YOLOv5

Pelatihan YOLOv5 dilakukan dengan memanfaatkan metode optimasi seperti SGD atau Adam. Model belajar melalui iterasi pada setiap batch data, di mana prediksi yang dihasilkan dihitung dengan nilai loss berdasarkan selisih dengan target sebenarnya. Scheduler digunakan untuk mengatur learning rate menggunakan cosine decay, yang membantu model untuk mencapai konvergensi dengan lebih stabil.

Validasi dan *Early Stopping*, Setelah setiap *epoch*, performa model divalidasi menggunakan dataset yang tidak termasuk dalam pelatihan. Metrik evaluasi seperti *precision*, *recall*, dan *mean Average Precision* (mAP) dihitung untuk mengukur kinerja model. Mekanisme *Early Stopping* diterapkan untuk menghentikan pelatihan ketika performa tidak meningkat dalam beberapa *epoch* berturut-turut, membantu model menghindari *overfitting*.

```
The dataset includes 5555 images.
License-plate and vehicle are annotated in YOLO v5 PyTorch format.

The following pre-processing was applied to each image:
* Auto-orientation of input data (with EXIF-orientation stripping)
* Resize to 640x640 (Stretch)

The following augmentation was applied to create 3 versions of each source image:
* 50% probability of horizontal flip
* Random rotation of between -10 and +10 degrees
```

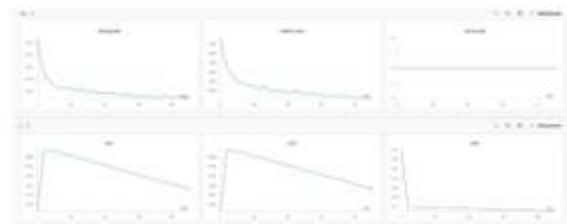
Gambar 4. Hasil Evaluasi Model YOLOv5 pada Dataset Validasi

Model terbaik yang mencapai performa tertinggi dalam mAP disimpan sebagai best.pt, sementara model terakhir disimpan sebagai last.pt. Penyimpanan model ini memungkinkan penggunaan model terbaik untuk inferensi atau kelanjutan pelatihan jika diperlukan. Berikut menunjukkan baris kode yang berfungsi untuk

mengimpor berbagai pustaka dan modul yang diperlukan dalam pengembangan sistem ALPR berbasis YOLOv5 dan *Google Vision* OCR. Setiap pustaka yang diimpor memiliki fungsi spesifik dalam mendukung deteksi dan pengenalan karakter pada plat nomor kendaraan

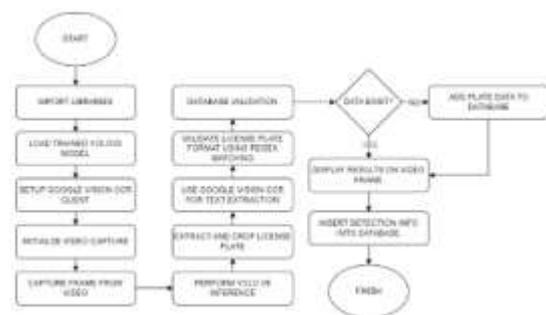
```
import torch
import cv2
from google.cloud import vision
from google.oauth2 import service_account
import io
import os
import re
import warnings
import mysql.connector
from mysql.connector import errorcode
```

Gambar 5. Pustaka dan Modul yang Digunakan untuk Implementasi Sistem ALPR



Gambar 6. Grafik Metrik Evaluasi dan Parameter Pelatihan Model YOLOv5

Perancangan Sistem ALPR dengan YOLOv5 dan *Google Vision* OCR, Sistem ini terdiri dari dua komponen utama: deteksi plat nomor menggunakan YOLOv5 dan pengenalan karakter menggunakan *Google Vision* OCR. YOLOv5 mendeteksi posisi plat nomor dalam gambar, dan hasil deteksi kemudian dikirim ke *Google Vision* OCR untuk mengonversi karakter pada plat nomor menjadi teks.



Gambar 7. Alur Kerja Sistem ALPR

III. HASIL DAN PEMBAHASAN

Program ini memiliki tujuan utama untuk mendeteksi dan mengenali plat nomor kendaraan secara otomatis dengan menggunakan dua teknologi utama, yaitu YOLOv5 untuk deteksi objek dan *Google Vision* OCR untuk pengenalan teks. Setiap tahap dalam program ini dirancang untuk memproses plat nomor dari video secara real-time dan mencatatnya ke dalam database

jika plat tersebut memenuhi kriteria tertentu. Teknologi yang digunakan memungkinkan proses yang efisien dalam mendeteksi dan mengenali plat nomor kendaraan dengan akurasi tinggi, sehingga program ini sangat berguna dalam berbagai aplikasi seperti pengawasan lalu lintas, parkir otomatis, dan pelacakan kendaraan.



Gambar 8. Pemuatan Model YOLOv5 dengan Torch Hub

Pada tahap awal, program mengimpor library yang dibutuhkan untuk menjalankan keseluruhan proses. Library utama yang diimpor termasuk torch untuk pemrosesan model YOLOv5, cv2 (OpenCV) untuk mengelola video dan gambar, serta google.cloud.vision untuk terhubung dengan Google Vision OCR. Selain itu, program juga mengimpor mysql.connector untuk berinteraksi dengan basis data MySQL yang akan menyimpan data plat nomor kendaraan yang terdeteksi. Library tambahan lainnya seperti io, os, dan re digunakan untuk membantu dalam pengolahan file, pengecekan direktori, dan validasi format teks plat nomor menggunakan ekspresi reguler.



Gambar 9. onfigurasi Autentikasi dan Inisialisasi Google Vision API

Tahap berikutnya adalah memuat model YOLOv5 yang akan digunakan untuk mendeteksi objek plat nomor pada setiap frame dalam video. Model ini diload dari path lokal yang telah disiapkan (`path='C:/Users/risqu/Desktop/PLATE/yolov5/runs/train/exp6/weights/best.pt'`), di mana model ini telah dilatih khusus untuk mengenali objek plat nomor. Dengan meload model secara lokal, program dapat mengakses model yang sesuai dengan kebutuhan deteksi, misalnya model yang sudah dikustomisasi untuk mendeteksi plat nomor sesuai dengan format tertentu.



Gambar 10. Fungsi Validasi Format Plat Nomor

Setelah memuat model YOLOv5, program mengatur kredensial untuk terhubung dengan Google Vision OCR. Kredensial ini penting karena Google Vision OCR merupakan layanan berbasis cloud yang memerlukan autentikasi. Kredensial ini disimpan dalam file JSON yang disediakan oleh Google Cloud Platform, yang path-nya telah ditentukan pada variabel `key_path`. Program membaca file JSON ini untuk mengautentikasi dan membuat klien Google Vision API yang nantinya akan digunakan untuk memproses gambar dan melakukan OCR.



Gambar 11. Fungsi Ekstraksi Teks dari Gambar Plat Nomor Menggunakan Google Vision OCR

Untuk menyimpan gambar hasil cropping dari plat nomor yang terdeteksi, program membuat direktori bernama `cropped_plates`. Direktori ini digunakan untuk menampung semua gambar plat nomor yang berhasil dicrop dari frame video yang diambil secara real-time. Jika direktori ini belum ada, program akan otomatis membuatnya dengan perintah `os.makedirs('cropped_plates')`. Penyimpanan gambar di direktori khusus ini memudahkan pengelolaan gambar hasil deteksi dan memisahkan data plat nomor dari file-file lain yang mungkin berada dalam direktori kerja.

Salah satu fungsi utama dalam program ini adalah `detect_text`, yang berfungsi untuk mengenali teks pada plat nomor dengan menggunakan Google Vision OCR. Fungsi ini pertama-tama membuka file gambar plat nomor, membaca kontennya sebagai byte array, dan mengirimkan konten tersebut ke Google Vision OCR untuk diproses. Google Vision kemudian mengembalikan hasil deteksi teks berupa string. Untuk memastikan teks yang terdeteksi sesuai dengan format plat nomor yang diinginkan, program menggunakan regex (ekspresi reguler) untuk menyaring teks. Sebagai contoh, program dapat mengenali plat nomor sipil dengan format `XX YYYY ZZZ`, di mana X adalah huruf dan Y serta Z adalah angka atau huruf.

```
def detect_text(image_path):  
    # Load image and convert to grayscale  
    image = cv.imread(image_path, cv.IMREAD_GRAYSCALE)  
    # Detect edges using Canny edge detection  
    edges = cv.Canny(image, 1, 30, 3)  
    # Find contours in the edge image  
    contours, _ = cv.findContours(edges, cv.RETR_LIST, cv.CHAIN_APPROX_SIMPLE)  
    # Iterate over contours to find license plate regions  
    for contour in contours:  
        # Approximate the contour as a rectangle  
        x, y, w, h = cv.boundingRect(contour)  
        # Crop the license plate region from the original image  
        plate = image[y:y+h, x:x+w]  
        # Perform OCR on the license plate region  
        text = pytesseract.image_to_string(plate, config='--psm 6')  
        # Extract the license plate number (assuming 4 digits followed by 3 letters)  
        match = re.match(r'^([A-Z]{1,2})\d{4}([A-Z]{3})$', text)  
        if match:  
            # Return the license plate number and region coordinates  
            return match.group(1), (x, y, x+w, y+h)  
    return None
```

Gambar 12. Fungsi Pencarian Wilayah Berdasarkan Kode Plat Nomor

Selain fungsi OCR, program juga memiliki fungsi `is_valid_license_plate` untuk memvalidasi apakah teks yang dideteksi benar-benar sesuai dengan format plat nomor yang diinginkan. Fungsi ini menggunakan pola regex untuk mengecek apakah teks plat nomor mengikuti format tertentu, seperti `^([A-Z]{1,2} \d{4} [A-Z]{0,3})$`. Pola ini memastikan bahwa hanya plat nomor yang valid yang akan diproses lebih lanjut. Jika teks yang dideteksi tidak sesuai dengan pola ini, program akan mengabaikan plat nomor tersebut dan tidak menyimpannya ke dalam database.

```
def connect_mysql():  
    # MySQL connection details  
    host = 'localhost', Database = 'license_plate_db'  
    cursor = conn.cursor()  
    # Create table if not exists  
    create_table_query = '''  
    CREATE TABLE IF NOT EXISTS plates (  
        id INT AUTO_INCREMENT PRIMARY KEY,  
        plate_number VARCHAR(10) NOT NULL UNIQUE,  
        timestamp DATETIME NOT NULL  
    )  
    '''  
    cursor.execute(create_table_query)  
    conn.commit()  
    # Close connection  
    conn.close()  
    return cursor
```

Gambar 13. Koneksi dan Pembuatan Tabel di Database MySQL

Fungsi `get_plate_region` digunakan untuk menentukan lokasi asal plat nomor berdasarkan kode pada awal nomor plat. Fungsi ini terhubung dengan basis data *MySQL* yang menyimpan informasi mengenai kode plat nomor dan asal daerahnya. Saat sebuah plat nomor terdeteksi, program akan mengambil dua karakter pertama sebagai kode wilayah dan menggunakan fungsi ini untuk mencari nama daerah dan provinsi asal plat tersebut. Informasi ini berguna untuk memberikan data geografis pada plat nomor yang terdeteksi, terutama pada plat sipil yang memiliki kode wilayah khusus.

Untuk menyimpan hasil deteksi plat nomor, program terhubung dengan basis data *MySQL*. Dalam proses ini, program terlebih dahulu mencoba untuk membuat koneksi dengan server

database dan login dengan kredensial yang telah ditetapkan. Program juga membuat tabel bernama `plates` jika tabel tersebut belum ada. Tabel ini memiliki kolom `id`, `plate_number`, dan `timestamp`, di mana setiap entri menyimpan informasi plat nomor yang terdeteksi beserta waktu deteksinya. Dengan menyimpan data ini, program dapat melacak plat nomor yang telah terdeteksi dan mencegah duplikasi data.

```
# Video capture and processing  
cap = cv2.VideoCapture(0) # 0: default webcam  
while True:  
    ret, frame = cap.read()  
    if not ret:  
        break  
    # Detect license plate using YOLOv5  
    results = model(frame)  
    # Iterate over detected license plates  
    for data in results[0]:  
        # Bounding box of the license plate  
        x1, y1, x2, y2 = data[0], data[1], data[2], data[3]  
        # Crop the license plate region  
        plate = frame[y1:y2, x1:x2]  
        # Detect license plate number using OCR  
        text = pytesseract.image_to_string(plate, config='--psm 6')  
        # Extract the license plate number  
        match = re.match(r'^([A-Z]{1,2})\d{4}([A-Z]{0,3})$', text)  
        if match:  
            # License plate number and region coordinates  
            plate_number = match.group(1) + match.group(2) + match.group(3)  
            # Region coordinates (x1, y1, x2, y2)  
            region = (x1, y1, x2, y2)  
            # Save the license plate number and region to the database  
            cursor.execute('INSERT INTO plates (plate_number, timestamp) VALUES (%s, %s)',  
                          (plate_number, datetime.now()))  
            # Print the license plate number and region  
            print(f'Detected License Plate: {plate_number}')  
            # Show the license plate region on the original frame  
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)  
            # Display the original frame with the license plate region highlighted  
            cv2.imshow('License Plate Region', frame)
```

Gambar 14. Proses Deteksi dan Pengenalan Plat Nomor Menggunakan YOLOv5 dan Google Vision OCR

Setelah semua persiapan selesai, program beralih ke tahap deteksi real-time. Program memulai video capture dari kamera (dalam contoh ini, kamera default diindikasikan oleh `cap = cv2.VideoCapture(0)`). Program membaca setiap frame dari video dan menggunakan model YOLOv5 untuk mendeteksi objek di dalamnya. Jika objek plat nomor terdeteksi, program akan menyimpan posisi koordinat objek tersebut dan memotong gambar plat nomor dari frame sesuai dengan koordinat yang diberikan.

Setelah plat nomor dicrop, gambar ini disimpan dengan nama yang sesuai dalam direktori `cropped_plates`. Gambar yang telah disimpan kemudian diproses dengan fungsi

detect_text untuk mengenali teks pada plat nomor tersebut. Hasil OCR ini kemudian dicek apakah valid menggunakan fungsi is_valid_license_plate. Jika teks yang dihasilkan sesuai dengan format plat nomor sipil, maka program melanjutkan proses penyimpanan data.

Jika plat nomor yang terdeteksi adalah plat sipil, program juga akan mengidentifikasi wilayah asalnya dengan memanfaatkan fungsi get_plate_region. Fungsi ini mencari informasi kode wilayah pada basis data dan mengembalikan nama daerah serta provinsi asal plat nomor. Informasi ini ditampilkan bersama teks plat nomor pada video, memberikan konteks tambahan mengenai asal plat nomor tersebut. Hal ini bermanfaat untuk keperluan pelacakan dan analisis data lebih lanjut.

Untuk menghindari duplikasi, program memeriksa apakah plat nomor yang terdeteksi sudah ada di database. Jika plat nomor sudah ada, program tidak akan menambahkannya kembali tetapi tetap menampilkan informasi wilayah di layar. Jika plat nomor belum ada di database, program akan menambahkannya dengan timestamp waktu deteksi saat ini. Langkah ini memastikan bahwa data dalam database tetap unik dan tidak terjadi pengulangan yang tidak perlu.

Setelah plat nomor valid tersimpan di database, program menampilkan bounding box dan informasi terkait pada video secara real-time. Setiap plat nomor yang terdeteksi akan diberikan bounding box berwarna hijau untuk menandai posisinya pada layar, serta teks yang menunjukkan informasi wilayah dan validasi plat. Visualisasi ini memberikan feedback langsung kepada pengguna mengenai plat nomor yang telah terdeteksi dan divalidasi oleh sistem.

Jika plat nomor yang terdeteksi tidak valid, program mengabaikan teks OCR tersebut dan tidak melakukan proses lebih lanjut. Hal ini memungkinkan program untuk hanya menyimpan data yang relevan dan valid. Selain itu, program juga menampilkan pesan "Ignored Invalid License Plate" di terminal untuk setiap plat nomor yang tidak sesuai format.

```
# Exit on pressing 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()
```

Gambar 15. Mengakhiri Program dan Menutup Jendela Video

Program akan terus berjalan dan memproses setiap frame dari video sampai pengguna menekan tombol 'q' pada keyboard. Saat tombol 'q' ditekan, program akan keluar dari loop dan menghentikan proses video capture. Kamera akan dirilis, dan semua jendela tampilan yang terkait akan ditutup untuk mengakhiri program.

```
# Close the database connection
cursor.close()
cnx.close()
```

Gambar 16. Menutup Koneksi Database

Pada tahap akhir, program memastikan koneksi dengan database ditutup dengan benar. Penutupan koneksi database ini penting untuk mencegah masalah akses di masa mendatang dan memastikan data tetap aman.

Setelah sistem deteksi dan pengenalan plat nomor kendaraan berhasil diimplementasikan menggunakan model YOLOv5, dan Google Vision OCR, dilakukan pengujian untuk menilai kinerja serta akurasi. Hasil pengujian mencakup efektivitas deteksi objek plat nomor, ketepatan ekstraksi teks menggunakan OCR, dan waktu pemrosesan. Berikut ini disajikan hasil pengujian yang diperoleh dari beberapa kondisi berbeda, termasuk variasi pencahayaan dan sudut pandang.

Pengujian model deteksi plat nomor dan Optical Character Recognition (OCR) dengan Google Vision API dalam aplikasi ini dirancang untuk mendeteksi dan mengenali teks pada plat nomor kendaraan dengan akurat. Sistem ini memanfaatkan model YOLOv5 untuk deteksi objek plat nomor serta Google Vision API untuk mengenali teks pada plat nomor tersebut. Aplikasi ini dibangun menggunakan Flask sebagai framework untuk memungkinkan pengujian dilakukan melalui antarmuka web yang interaktif. Dengan demikian, pengguna dapat mengunggah gambar dan melihat hasil deteksi dan ekstraksi teks secara langsung.

Pengujian dimulai dengan mengimpor pustaka-pustaka yang diperlukan, termasuk torch untuk pemanggilan model YOLOv5, cv2 untuk pengolahan citra, serta google.cloud.vision untuk menjalankan fungsi OCR menggunakan Google Vision API. Selain itu, pustaka random digunakan untuk memilih gambar secara acak dalam uji coba, sedangkan shutil membantu dalam pengelolaan file dengan menyalin gambar ke direktori yang diinginkan. Setelah pustaka diimpor, Flask diinisialisasi untuk menjalankan aplikasi web yang berfungsi sebagai antarmuka antara pengguna dan model deteksi.

Langkah berikutnya adalah melakukan inisialisasi model deteksi YOLOv5 dengan memuat bobot model yang telah dilatih untuk mendeteksi objek spesifik, yaitu plat nomor kendaraan. File bobot model disimpan dalam direktori khusus dan dipanggil menggunakan pustaka torch. Selain itu, Google Vision API dikonfigurasi dengan autentikasi melalui kredensial yang disimpan dalam file JSON. Kredensial ini diperlukan untuk mengakses layanan Google Vision API dan memungkinkan aplikasi melakukan OCR pada gambar.

Dalam proses pengujian ini, beberapa direktori penting disiapkan, yaitu `cropped_plates` untuk menyimpan gambar plat nomor yang sudah terpotong, dan `evaluasi_tes` untuk menyimpan gambar-gambar uji yang akan diproses. Aplikasi secara otomatis memeriksa apakah direktori tersebut sudah ada atau belum, dan membuatnya jika belum tersedia. Hal ini penting untuk menjaga agar struktur penyimpanan gambar tetap rapi dan mudah diakses.

Fungsi `detect_text` merupakan komponen utama yang menggunakan Google Vision API untuk mengekstraksi teks dari gambar plat nomor. Fungsi ini membuka file gambar dan mengirimkannya ke Google Vision API untuk dideteksi teksnya. Jika teks terdeteksi, hasilnya akan difilter untuk menyesuaikan dengan pola format plat nomor tertentu. Misalnya, format plat nomor sipil biasanya terdiri dari huruf dan angka dalam pola khusus, dan pola ini digunakan untuk menyaring hasil OCR agar lebih relevan.

Pada halaman utama aplikasi, route index diatur untuk menampilkan hasil deteksi. Route ini mengambil 10 gambar secara acak dari direktori sumber yang sudah ditentukan, lalu menyalinnya ke direktori `evaluasi_tes`. Setelah gambar dipilih, aplikasi memulai proses deteksi plat nomor menggunakan model YOLOv5. Model ini membaca gambar dan menentukan lokasi plat nomor pada gambar, lalu memotong bagian tersebut dan menyimpannya di direktori `cropped_plates` sebagai gambar baru.

Setiap gambar yang telah terpotong kemudian diproses oleh fungsi `detect_text` untuk mengekstraksi teks plat nomor. Jika OCR berhasil mendeteksi teks yang sesuai dengan pola plat nomor, hasilnya dicatat. Informasi yang disimpan meliputi path gambar asli, path gambar hasil cropping, dan teks plat nomor yang terdeteksi. Semua hasil ini kemudian dikirim ke template HTML untuk ditampilkan.

Pada sisi frontend, template HTML yang disiapkan menggunakan struktur tabel untuk

menampilkan hasil deteksi. Tabel ini menampilkan gambar asli, gambar plat nomor yang sudah di-crop, dan teks plat nomor yang berhasil dideteksi oleh sistem. Tampilan tabel yang sederhana namun informatif ini memudahkan pengguna untuk melihat perbandingan antara gambar asli dan hasil deteksi serta OCR, sehingga hasil uji menjadi lebih mudah dipahami.

Pengujian keseluruhan mencakup berbagai elemen penting, mulai dari akurasi deteksi plat nomor oleh YOLOv5, keberhasilan cropping plat nomor, hingga ketepatan OCR dalam mengenali teks pada plat nomor. Hasil pengujian akan menunjukkan sejauh mana model YOLOv5 dan Google Vision API mampu bekerja bersama untuk mendeteksi dan mengenali plat nomor kendaraan dengan tingkat akurasi yang diharapkan.

Dalam tahap pengujian, digunakan diambil 5 sampel random dari direktori `dataset\valid\images`, yang dimana direktori ini terdapat 432 gambar.

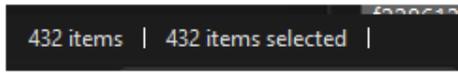


Gambar 17. direktori `dataset\valid\images`

Penjelasan program ini diawali dengan import pustaka: pustaka `random` digunakan untuk memilih gambar secara acak, sementara `shutil` berfungsi untuk menyalin file. Program kemudian melakukan inisialisasi Flask App, dengan membuat instance Flask serta menginisialisasi model YOLOv5 dan Google Vision API untuk mendeteksi objek dan teks pada gambar. Selanjutnya, program akan membuat direktori `cropped_plates` dan `evaluasi_tes` jika direktori tersebut belum ada, sehingga gambar yang akan diproses dapat disimpan dengan tertata.

Dalam proses deteksi teks, fungsi `detect_text` menggunakan Google Vision API untuk mendeteksi teks pada gambar yang diunggah. Program kemudian menetapkan route untuk halaman utama yang berfungsi memilih sejumlah gambar secara acak dari direktori `C:/.../yolov5/dataset/valid/images` dan menyalinnya ke direktori `evaluasi_tes`. Pada

gambar-gambar di direktori evaluasi tes ini, deteksi plat nomor dilakukan, dan hasilnya kemudian ditampilkan di halaman web. Untuk template HTML, program menampilkan gambar asli, gambar cropped plate, dan hasil deteksi plat nomor dalam bentuk tabel, sehingga pengguna dapat melihat hasilnya dengan jelas dan mudah.



Gambar 18. Pengujian Pertama, akurasi 100%



Gambar 19. Pengujian kedua 100%



Gambar 20. Pengujian ketiga 100%



Gambar 21. Pengujian keempat 100%



Gambar 22. Pengujian kelima 100%

Dalam pengujian digunakan 25 sampel dari 432 gambar, dan memperoleh akurasi sebesar 100%, bahkan dapat mendeteksi secara low light dan tampilan mirror.

IV. SIMPULAN DAN SARAN

A. Simpulan

Dalam hal ini, implementasi sistem ALPR menggunakan YOLOv5 dan Google Vision OCR telah terbukti efektif dalam melakukan tugasnya, yakni mendeteksi dan mengenali plat nomor Indonesia secara otomatis. Dengan YOLOv5 terbukti mampu mendeteksi plat nomor pada gambar dengan akurat dalam waktu yang singkat. Sementara itu, Google Vision OCR juga terbukti mampu memproses karakter plat nomor dengan tingkat kesalahan yang sangat rendah, bahkan dalam kondisi lingkungan yang beragam, termasuk intensitas cahaya yang tinggi dan rendah serta sudut pengambilan gambar.

Selanjutnya, tugas evaluasi algoritma menunjukkan bahwa model yang dikembangkan konsisten dalam mendeteksi plat nomor, sebagaimana ditunjukkan dalam hasil deteksi algoritma pada video streaming. Terbukti juga dengan penyimpanan plat nomor ke database *MySQL* yang valid dan sejalan dengan waktu deteksi. Tetapi model harus ditingkatkan pada deteksi plat nomor yang mengalami rolling. Dari sisi operasionalitas, hasil pengujian sistem ALPR menunjukkan bahwa sistem mampu menyimpan semua informasi lebih akurat dan terstruktur dalam database *MySQL*.

Secara keseluruhan, dari data dan informasi yang telah didiskusikan, pada implementasi ini, sistem ALPR ini bias mempunyai banyak potensi dalam pengaplikasian ALPR pada pengawasan lalu lintas, sistem parkir atau sebagainya di mana membutuhkan informasi dengan waktu yang real-time sempurna atau deteksi plat nomor dengan akurat yang tak terkalahkan.

B. Saran

Pembahasan terkait penelitian ini masih sangat terbatas dan membutuhkan banyak masukan, saran untuk penulis selanjutnya adalah mengkaji lebih dalam dan secara komprehensif tentang

DAFTAR RUJUKAN

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

- Hidayatullah, P., Syakrani, N., Suhartini, I., & Muhlis, W. (2012). Optical Character Recognition Improvement for License Plate Recognition in Indonesia. *2012 UKSim-AMSS 6th European Modelling Symposium*, pp. 236-249. <https://doi.org/10.1109/EMS.2012.95>.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*.
- Jocher, G. (2020). YOLOv5. Retrieved from <https://github.com/ultralytics/yolov5>.
- Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.
- Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. <https://doi.org/IJCNN.2018.8489629>.
- Lin, T. Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., ... Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.
- Salma, G., & Putri, D. U. K. (2023). Deteksi Objek dan Pengenalan Karakter Plat Nomor Kendaraan dengan Metode Deep Learning. *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, Vol.x, No.x, pp. 1-5.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60.
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How Transferable Are Features in Deep Neural Networks?. In *Advances in Neural Information Processing Systems (NIPS)*.