

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

2.1.1 Electronic Toll Collection System Based On Computer Vision [8]

Penelitian “Electronic Toll Collection System Based On Computer Vision” yang diteliti oleh K. Manasa dan V. Madhurima ini merupakan salah satu penelitian yang mengimplementasikan teknologi *Computer Vision* untuk sistem deteksi kendaraan dan plat nomor di jalan tol. Sistem yang diusulkan adalah *framework* berbasis *Computer Vision* yang cocok untuk lokasi dengan kendaraan bergerak dan pengecekan jalan. Sebuah kamera akan menangkap gambar kendaraan yang melewati tikungan tol kemudian kendaraan dideteksi melalui kamera. *Framework* yang diusulkan tergantung pada identifikasi kendaraan Menggunakan *library* Open CV lalu dieksekusi menggunakan sistem operasi Linux.

Menurut peneliti dari jurnal ini, waktu reaksi lebih cepat dan lebih terjangkau, lebih praktis, dan lebih efektif daripada beberapa *framework* lainnya. Terlebih lagi, untuk penggunaan ini. Begitu juga dengan penghitungan jumlah kendaraan yang melewati tikungan tol. Sistem juga dapat mengidentifikasi kendaraan ringan dan berat. Data ini diteruskan ke *Raspberry Pi* yang memiliki server web yang sudah diatur.

Ketika *raspberry pi* mendeteksi mobil, kemudian sistem akan memberikan data beserta harga tol yang harus dibayar. Pada penelitian ini sistem berhasil untuk diimplementasikan.

Beberapa poin penting yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

- Penelitian ini menggunakan sistem operasi Linux yang dianggap memiliki reaksi yang lebih cepat dan responsive.

- Penggunaan *library* OpenCV dapat menjadi pilihan karena sudah banyak digunakan dalam berbagai aplikasi yang melibatkan *Computer Vision*.

2.1.2 Computer Vision Based Vehicle Detection for Toll Collection System Using Embedded Linux [9]

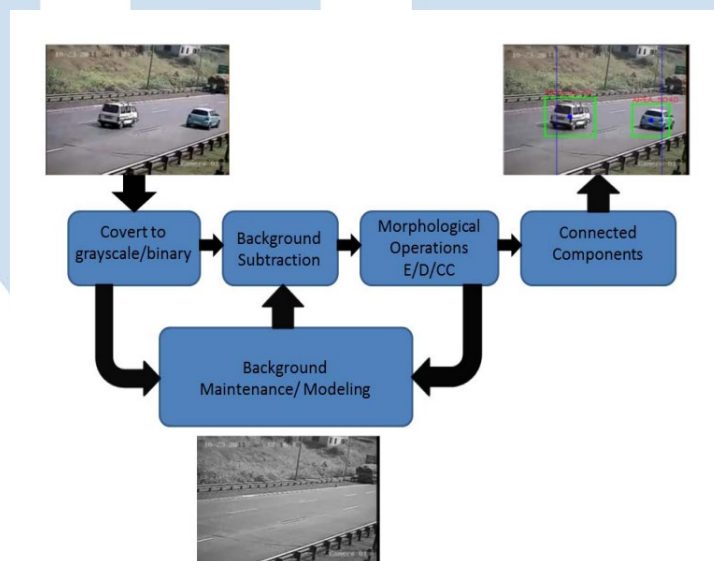
Penelitian ini menyajikan tinjauan singkat tentang *Toll Collection System* yang ada di India, kelebihan dan kekurangannya dan juga bertujuan untuk merancang dan mengembangkan *Toll Collection System* baru yang efisien yang akan menjadi alternatif yang lebih murah dan baik di antara semua sistem lainnya. Sistem ini didasarkan pada deteksi kendaraan menggunakan *Computer Vision* yang menggunakan *library* OpenCV di platform *Embedded Linux*. Sistem ini dirancang dengan menggunakan *Embedded Linux development kit (Raspberry pi)*. Pada sistem ini, kamera menangkap gambar kendaraan yang melewati pintu tol sehingga kendaraan terdeteksi melalui kamera.

Sistem ini dibagi menjadi 3 tahap utama, yang pertama adalah tahap inisialisasi dimana sistem akan menginisialisasi kamera untuk memberikan data video untuk di analisis. Tahap kedua adalah *background subtraction* yang dilakukan pada gambar menggunakan algoritma *Kalman Filter*. Tahap ketiga adalah *vehicle detection*, di tahap ini sistem mendapatkan hasil dari tahap sebelumnya lalu beberapa operasi *morphological* dilakukan untuk mendeteksi kendaraan dan *men-tracking* kendaraan tersebut.

Setelah tahap deteksi, tahap selanjutnya adalah mengklasifikasikan kendaraan. Dalam penelitian ini, wilayah kendaraan yang terdeteksi yang digambar menggunakan fungsi *library* OpenCV yang kemudian diklasifikasikan sebagai kendaraan ringan atau berat. Peneliti dari penelitian ini membuat log rinci kendaraan yang memberikan informasi seperti berapa banyak kendaraan yang masuk dan keluar; Dari jumlah tersebut, berapa banyak kendaraan ringan dan berat dalam bentuk file teks dengan rincian tersebut.

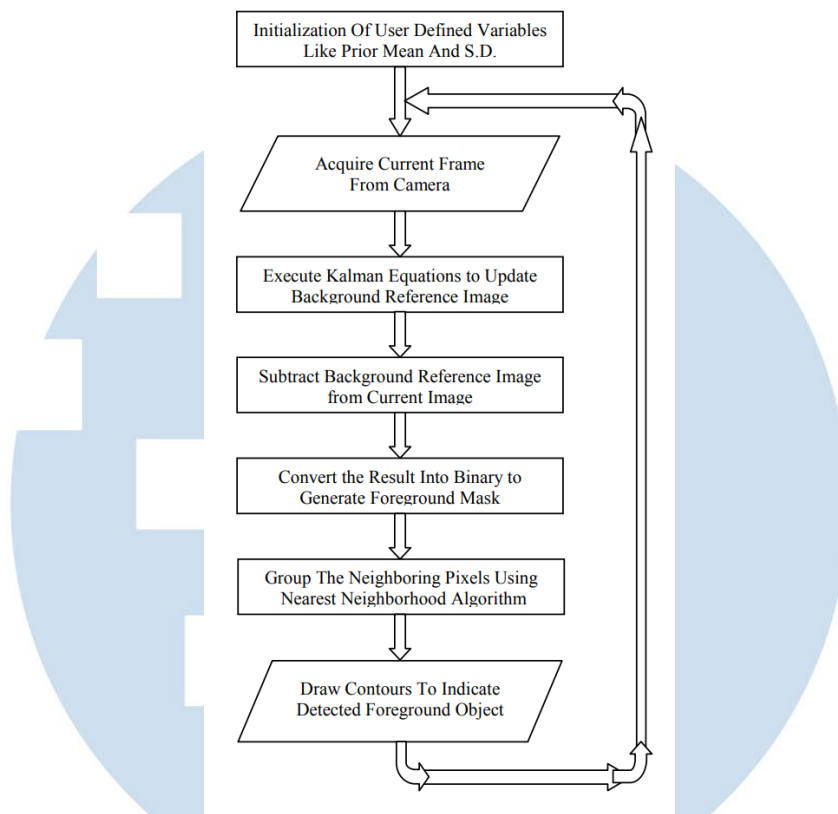
Beberapa poin penting yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

- Pengujian pada algoritma menunjukkan bahwa ambang batas varian antara latar depan dan latar belakang adalah parameter penting yang harus dicari.
- Penggunaan filter *Kalman* adalah algoritma yang sangat kuat karena cukup kuat dan efisien sehingga dapat diimplementasikan pada platform *embedded*.



Gambar 2.1 Arsitektur Sistem

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 2.2 Pipeline Pengaplikasian Kalman Filter

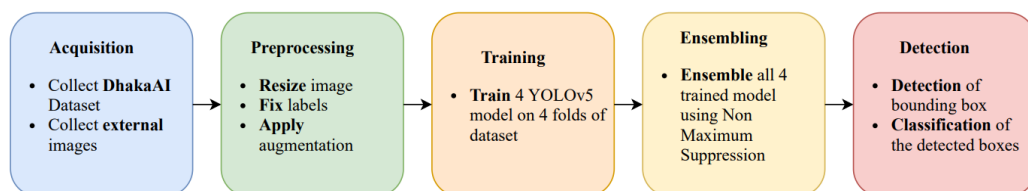
2.1.3 Densely-Populated Traffic Detection using YOLOv5 and Non-Maximum Suppression Ensembling [10]

Dalam penelitian ini, para peneliti mengusulkan metode yang dapat menemukan dan mengklasifikasikan objek kendaraan dari gambar kerumunan yang diberikan menggunakan YOLOv5. Kekurangan YOLO diselesaikan dengan 4 model berbeda. Model yang diusulkan berkinerja baik pada gambar yang diambil dari tampilan atas dan samping jalan di siang dan malam hari. Kinerja model yang diusulkan oleh peneliti diukur pada dataset Dhaka AI yang berisi gambar kendaraan berkerumun atau padat. Penelitian ini menunjukkan bahwa model yang dibuat oleh para peneliti mencapai $mAP@0,5$ sebesar 0,458 dengan waktu inferensi 0,75 detik yang mengungguli model terancang lainnya dalam hal kinerja. Oleh karena itu, model dapat diimplementasikan di jalan untuk deteksi lalu lintas waktu nyata yang dapat digunakan untuk pengendalian lalu lintas dan pengumpulan data.

Metode yang diusulkan terdiri dari 3 modul utama. Pertama, sistem akan memperoleh dan memproses dataset terlebih dahulu. Selama prapemrosesan, sistem menerapkan augmentasi, mengubah ukuran gambar menjadi bentuk yang seragam, dan membuat *training folds* dan pengujian. Kemudian, empat model berbeda dilatih dengan *training folds* yang berbeda ini. Setelah pelatihan, sistem menggabungkan model menggunakan *Non-Maximum Suppression* untuk inferensi akhir.

Beberapa poin penting yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

- Menggeneralisasi model pendeteksian objek menggunakan arsitektur deep learning, prasyaratnya adalah memiliki contoh pelatihan yang cukup untuk setiap kelas sehingga model dapat belajar dengan baik.
- *Framework* YOLO memiliki waktu inferensi yang jauh lebih sedikit dengan akurasi yang lebih baik. Di antara berbagai versi YOLO, peneliti memilih YOLOv5 karena arsitekturnya yang sederhana dibandingkan dengan model berbasis R-CNN.



Gambar 2.3 Pipeline sistem

2.1.4 Automatic License Plate Recognition in Real-World Traffic Videos Captured in Unconstrained Environment by a Mobile Camera [11]

Penelitian ini menyajikan teknik ALPR yang efisien berdasarkan *deep learning*, yang secara akurat melakukan tugas pengenalan plat nomor di lingkungan yang tidak dibatasi, bahkan ketika dilatih pada kumpulan data terbatas. Peneliti merekam video lalu lintas nyata di kota dan memberi label plat nomor dan karakter alfanumerik di plat nomor dalam *frame* yang berbeda untuk menghasilkan kumpulan data pelatihan dan pengujian. Teknik augmentasi data diterapkan untuk meningkatkan jumlah sampel pelatihan dan

pengujian. Peneliti dari penelitian ini menerapkan pendekatan pembelajaran transfer untuk melatih YOLOv5 yang baru saja dirilis.

Penelitian ini menyajikan model berbasis *deep learning* dua tahap untuk deteksi dan pengenalan plat nomor. Penelitian ini menunjukkan bahwa pendekatan pembelajaran transfer dengan model YOLOv5 yang ada dapat mencapai tingkat deteksi yang sangat tinggi untuk plat nomor dan alfanumeriknya dengan pelatihan minimal. Peneliti mengusulkan tahap kedua untuk pengenalan alfanumerik sebagai CNN khusus yang dilatih pada kumpulan data gambar depan dan belakang kendaraan yang dibuat khusus. Penelitian ini menunjukkan bahwa pendekatan dua tahap ini dapat mencapai akurasi yang sangat tinggi, dibandingkan dengan solusi deteksi dan pengenalan satu tahap. Peneliti dari penelitian ini menyajikan kumpulan data baru dari pelat yang ditangkap dalam lalu lintas perkotaan dunia nyata di Arab Saudi, dengan latar belakang dinamis yang tidak dibatasi.

Kumpulan data mewakili beberapa skenario yang sangat menantang yang tidak tercakup dalam kumpulan data yang tersedia. Penelitian ini menunjukkan bahwa, meskipun ukuran dataset pelatihan kecil, sistem dapat mencapai akurasi yang cukup tinggi dibandingkan dengan beberapa sistem ALPR canggih pada dataset benchmark yang ada, serta dataset kustom yang dibuat.

Beberapa poin penting yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

- Dengan membuat model kustom untuk aplikasi tertentu, dapat meningkatkan akurasi deteksi objek yang ingin dideteksi.
- Pengklasifikasian setiap alfanumerik merupakan tugas yang menantang bagi YOLOv5 dengan dataset pelatihan yang terbatas, dan hanya dapat mencapai akurasi 36% pada dataset penelitian ini.

2.2 Tinjauan Teori

2.2.1 Computer Vision

Computer Vision adalah sebuah istilah kombinasi antara *image processing* dan *pattern recognition*, pengembangan bidang ini dilakukan dengan mengadaptasi kemampuan penglihatan manusia dalam mengambil informasi. Sebagian besar tugas dalam *computer vision* terkait dengan proses memperoleh informasi tentang peristiwa atau deskripsi, dari adegan input (gambar digital) dan fitur ekstraksi. Metode yang digunakan untuk memecahkan masalah dalam visi komputer bergantung pada domain aplikasi dan sifat data yang dianalisis. Perkembangan *computer vision* sangat bergantung pada sistem teknologi komputer, baik tentang peningkatan kualitas gambar maupun pengenalan atau pendeteksi gambar. Tujuan utama Computer Vision adalah untuk membuat model dan mengekstrak data dan informasi dari gambar, sedangkan *Image Processing* adalah tentang menerapkan transformasi komputasi untuk gambar, seperti penajaman, kontras, dan hal-hal lainnya yang berkaitan dengan gambar [12].

Secara fungsional *computer vision* dan indera penglihatan manusia adalah sama, dengan tujuan untuk menginterpretasikan data spasial, yaitu data yang terindeks lebih dari satu dimensi. Namun, *computer vision* tidak dapat diharapkan untuk mereplikasi seperti mata manusia. Hal ini disebabkan sistem computer vision memiliki kinerja dan fungsi yang terbatas dibandingkan dengan mata manusia. Meskipun banyak ahli telah mengusulkan area luas teknik visi komputer untuk meniru mata manusia, namun, dalam banyak kasus, ada keterbatasan kinerja sistem visi komputer [13].

2.2.2 Object Detection

Object Detection adalah sebuah proses yang bertujuan untuk menemukan dan mengklasifikasikan objek yang ada dalam satu gambar, dan melabelinya dengan kotak pembatas untuk menunjukkan tingkat kepercayaan dari algoritma *Object Detection* terhadap benda yang dideteksi [14]. Model *objection detection* berbasis *deep learning* biasanya memiliki dua bagian.

Encoder yang akan mengambil gambar sebagai input dan menjalankannya melalui serangkaian blok dan *layer* yang akan mempelajari dan mengekstrak fitur statistik yang digunakan untuk mencari dan memberi label pada objek. Output dari encoder kemudian diteruskan ke decoder, yang memprediksi *bounding box* dan label untuk setiap objek [15]. Beberapa contoh model *object detection* yaitu, CNN, Faster R-CNN, YOLO.

2.2.3 Optical Character Recognition

Optical Character Recognition (OCR) adalah proses yang memungkinkan sistem tanpa campur tangan manusia mengidentifikasi skrip atau abjad yang ditulis ke dalam komunikasi verbal pengguna. OCR telah berkembang menjadi individu dari aplikasi pengetahuan yang berkembang pesat di bidang deteksi pola dan kecerdasan buatan [16]. OCR menganalisis pola terang dan gelap yang membentuk huruf dan angka untuk mengubah gambar yang dipindai menjadi teks. Sistem OCR perlu mengenali karakter dalam berbagai font, jadi beberapa aturan diterapkan untuk membantu sistem mencocokkan apa yang dilihatnya dalam gambar dengan huruf atau angka yang tepat, OCR memiliki beberapa kelemahan salah satunya ketika berhadapan dengan karakter-karakter dengan font yang berbeda [17].

2.2.4 YOLO

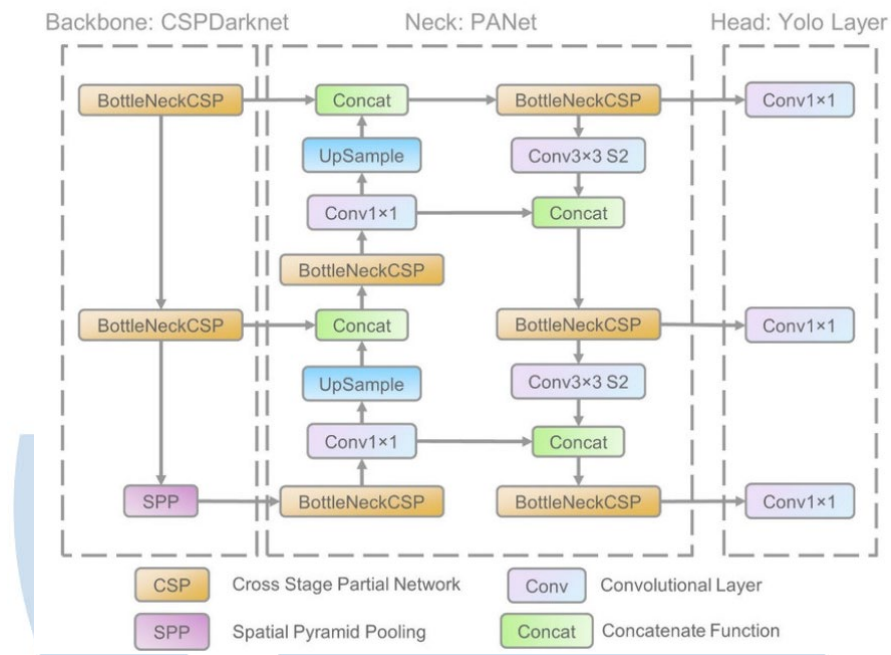
YOLO adalah singkatan dari *You Only Look Once*. Pendekatan ini menggunakan *single neural network* untuk memproses seluruh gambar, kemudian memisahnya menjadi beberapa bagian dan memprediksi *bounding box* dan probabilitas untuk setiap komponen. *Bounding box* ini diberi bobot dengan probabilitas yang diharapkan. Metode "*You Only Look Once*" pada gambar dalam arti bahwa itu membuat prediksi setelah hanya satu *forward propagation* yang dijalankan melalui *neural network*. Kemudian mengirimkan item yang terdeteksi setelah *non-max suppression* (yang memastikan bahwa algoritma deteksi objek hanya mengidentifikasi setiap

objek satu kali). Saat ini YOLO sudah mengeluarkan versi ke lima (5) atau biasa dikenal dengan YOLOv5 dari algoritma yang dibuat oleh ultralytics ini, YOLOv5 memiliki beberapa keunggulan dibandingkan versi pendahulunya yaitu YOLOv4, seperti; diperkirakan YOLOv5 memiliki ukuran algoritma yang lebih kecil dari pada YOLOv4 yaitu 27MB (*Mega Bytes*) dibandingkan dengan 244MB. Diperkirakan juga YOLOv5 memiliki kecepatan deteksi yang lebih cepat dibandingkan dengan YOLOv4 yaitu 140 FPS (*Frames Per Second*) dibandingkan dengan 50 FPS. YOLOv5 memiliki tingkat keakuratan yang lebih baik sedikit dibandingkan dengan versi pendahulunya YOLOv4, yaitu, 0.895 mAP (mean Average Precision) dibandingkan dengan 0.892 mAP [18]. Arsitektur dari YOLOv5 dibagi menjadi 3 bagian yaitu, *Backbone*, *Neck*, dan *Head*.

Model *Backbone* sebagian besar digunakan untuk mengekstrak fitur utama dari gambar input. CSP (*Cross Stage Partial Networks*) digunakan sebagai *Backbone* di YOLO v5 untuk mengekstrak fitur yang kaya akan karakteristik yang berguna dari gambar *input* [18].

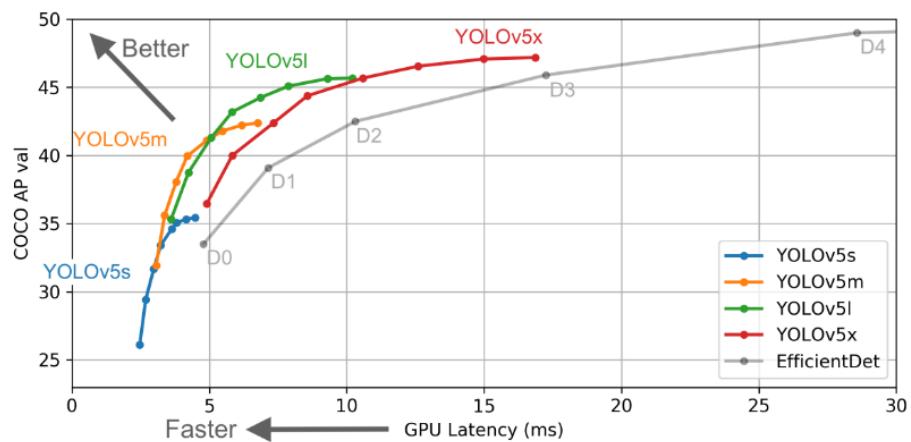
Model *Neck* sebagian besar digunakan untuk membuat *feature pyramid*. *Feature pyramid* membantu model dalam menggeneralisasi dengan sukses dalam hal penskalaan objek. Ini membantu dalam mengidentifikasi objek yang sama dalam berbagai ukuran dan skala. *Feature pyramid* cukup bermanfaat dalam membantu model untuk bekerja secara efektif pada data yang sebelumnya tidak terlihat. Model lain, seperti FPN, BiFPN, dan PANet, menggunakan berbagai macam pendekatan *feature pyramid*. PANet digunakan sebagai leher di YOLO v5 untuk mendapatkan *feature pyramid* [18].

Model *Head* sebagian besar bertanggung jawab untuk langkah deteksi akhir. Ini menggunakan *anchor box* untuk membangun vektor keluaran akhir dengan probabilitas kelas, skor objektivitas, dan *bounding box* [18].



Gambar 2.4 Arsitektur YOLOv5

Dibandingkan dengan model deteksi objek lainnya seperti Faster RCNN, YOLOv5 unggul dalam hal *Inference Speed*, dapat mendeteksi objek kecil atau jauh, hampir tidak ada *bounding box* yang saling tumpang dan dapat mendeteksi objek secara akurat di tempat yang ramai. Dalam hal performa YOLOv5 lebih unggul dibandingkan dengan Faster RCNN ResNet50 dengan hasil rata-rata 2.5 kali lebih cepat dengan memiliki performa deteksi objek kecil yang lebih baik dan juga hampir tidak ada *bounding box* yang saling tumpang [19].



Gambar 2.5 Perbandingan YOLOv5 dengan EfficientDet

2.2.5 Python

Python adalah bahasa pemrograman komputer yang sering digunakan untuk membangun situs web dan perangkat lunak, mengotomatisasi tugas, dan melakukan analisis data. Python adalah bahasa pemrograman *general-purpose*, artinya dapat digunakan untuk membuat berbagai macam program yang berbeda dan tidak khusus untuk suatu masalah tertentu. Fleksibilitas ini, bersama dengan tingkat kesulitan yang rendah membuat bahasa pemrograman ini cocok untuk pemula, menjadikannya salah satu bahasa pemrograman yang paling banyak digunakan saat ini.

Python umumnya digunakan untuk mengembangkan situs web dan perangkat lunak, otomatisasi tugas, analisis data, dan visualisasi data. Karena relatif mudah dipelajari, Python telah diadopsi oleh banyak non-programmer seperti akuntan dan ilmuwan, untuk berbagai tugas sehari-hari, seperti mengatur keuangan [20].

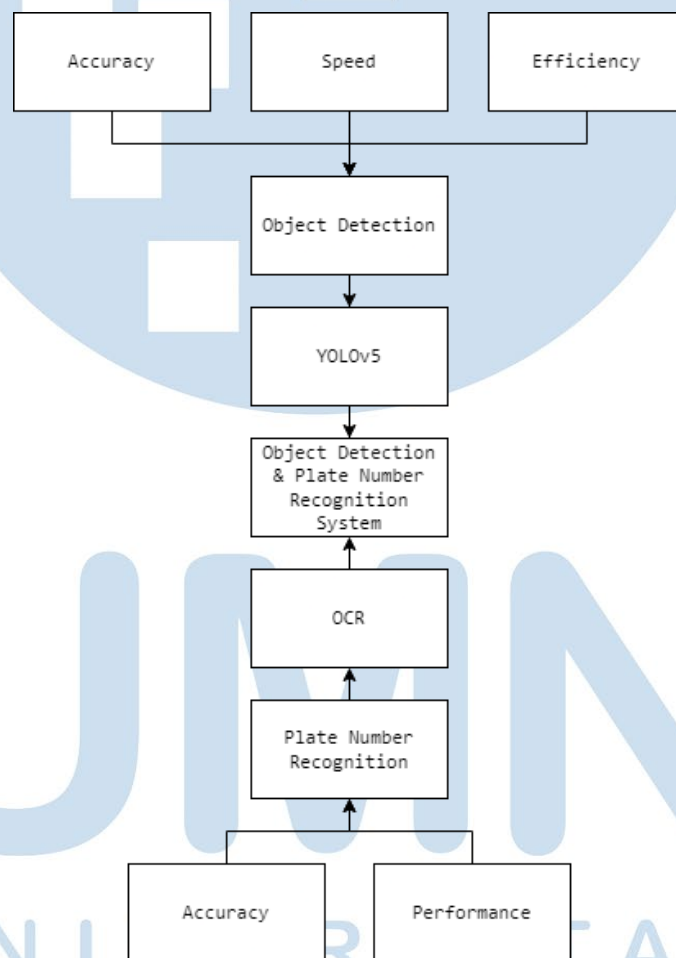
Python dirancang agar mudah dibaca, dan memiliki beberapa kesamaan dengan bahasa Inggris dengan pengaruh dari matematika. Python menggunakan baris baru untuk menyelesaikan perintah, berbeda dengan bahasa pemrograman lain yang sering menggunakan titik koma atau tanda kurung. Python bergantung pada *indentation*, menggunakan spasi, untuk mendefinisikan sebuah *scope*; seperti *scope* loop, fungsi dan kelas. Bahasa pemrograman lain sering menggunakan kurung kurawal untuk tujuan ini [21].

2.3 Summary

Berdasarkan studi pustaka yang telah dilakukan oleh penulis, maka rancangan sistem yang akan dibuat oleh penulis adalah sebagai berikut:

- Sistem akan menggunakan bahasa pemrograman Python, karena python merupakan salah satu bahasa pemrograman yang banyak dipakai untuk membuat perangkat lunak yang diaplikasikan dalam bidang *computer vision*. Bahasa pemrograman python juga digunakan pada penelitian [6], [7], dan [9].

- Sistem akan menggunakan model YOLOv5 sebagai model untuk mendeteksi kendaraan, karena YOLOv5 merupakan model *pre-trained* paling akurat dan kecepatan deteksi yang lebih cepat dibandingkan dengan YOLO versi sebelumnya pada saat penelitian ini dilakukan. YOLOv5 juga digunakan pada penelitian [9].
- Sistem akan menggunakan OCR (*Optical Character Recognition*) sebagai metode untuk mengekstraksi karakter-karakter yang ada di plat nomor kendaraan yang kemudian diubah menjadi teks.



Gambar 2.6 Diagram *state-of-the-art*

Berdasarkan diagram *state-of-the-art* diatas, dapat disimpulkan bahwa teknologi *Object Detection* khususnya model YOLOv5 saat ini memiliki tingkat

akurasi, kecepatan dan efisiensi dalam mendeteksi objek lebih baik dari sebelumnya, hal-hal tersebut merupakan aspek yang membuat model *object detection* ini sukses dan dipakai oleh banyak orang.

Teknologi OCR (*Optical Character Recognition*) adalah sebuah proses mengubah teks yang berisi gambar menjadi karakter yang dapat dibaca oleh komputer untuk diedit, dihitung, dan dianalisis. Saat ini teknologi OCR digunakan di banyak bidang perbankan, asuransi, hukum, dan layanan kesehatan. Aspek-aspek yang sangat penting dalam teknologi OCR yaitu, keakuratan sistem OCR dalam mengubah teks yang berisi gambar menjadi karakter dan performa sistem dalam melakukan proses mengubah teks yang berisi gambar menjadi karakter.

A large, light blue watermark logo of Universitas Multimedia Nusantara (UMMN) is centered on the page. It features a stylized 'U' and 'M' inside a circle, with a grid pattern overlaid.

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA