JTERA (Jurnal Teknologi Rekayasa), Vol. 7, No. 2, Desember 2022, Hal. 189-200
Terakreditasi "**Peringkat 3"** oleh Kemenristek/BRIN, Nomor SK: 85/M/KPT/2020
DOI: 10.31544/jtera.v7.i2.2022.189-200
p-ISSN 2548-737X
e-ISSN 2548-8678

Sistem Keamanan Gerbang Parkir Menggunakan Algoritma YOLO (*You Only Look Once*) dan *Face Recognition*

Tata Supriyadi[#], Kemal Taufik Fikri
Jurusan Teknik Elektro, Politeknik Negeri Bandung
Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, Jawa Barat 40559, Indonesia
[#]tata.supriyadi@polban.ac.id

Abstrak

Saat ini, sistem parkir masih menggunakan sistem parkir yang bersifat manual, dimana proses keluar masuknya kendaraan masih menggunakan pengecekan STNK atau karcis sehingga memungkinkan resiko kesalahan manusia dan rentan akan keamanan. Agar masalah ini dapat dikurangi, maka pada penelitian ini dibuatlah suatu sistem parkir cerdas pada *barrier gate* yang mana mampu mengidentifikasi wajah pengendara dan plat nomornya pada saat akan memasuki area parkir. Pada penelitian kali ini digunakan kecerdasan buatan menggunakan algoritma *You Only Look Once* (YOLO) untuk identifikasi dan pengenalan karakter plat nomor serta *library face recognition* untuk mengekstrak fitur pada wajah pengendara. Sistem berisi bagian yang mengontrol pembukaan gerbang keamanan pada bagian pintu keluar. Apabila wajah pengendara pada pintu keluar tidak cocok dengan pintu keluar maka sistem tidak mengizinkan gerbang keamanan untuk dibuka. Berdasarkan hasil penelitian yang telah dilakukan terhadap 372 gambar deteksi plat dan 169 gambar untuk pengenalan karakter didapat tingkat keberhasilan deteksi plat nomor sebesar 91,86% dengan *average* IOU 74,15% dan pengenalan karakter mencapai 95,8% dengan average IOU mencapai 81,25%. Adapun total total waktu komputasi sistem selama 4,85 detik pada Google Colabs dan 37,69 detik menggunakan Jetson Nano.

Kata kunci: parkir cerdas, kecerdasan buatan , YOLO, *face recognition, barrier gate*

Abstract

Currently, the parking system still uses a manual parking system, where the entry and exit process for vehicles still uses vehicle registration or ticket checking, which allows for the risk of human error and vulnerability to security. In order to reduce this problem, this research created an intelligent parking system at the barrier gate which is able to identify the driver's face and the number plate when entering the parking area. In this study, artificial intelligence was used using the You Only Look Once (YOLO) algorithm to identify and recognize license plate characters and the face recognition library to extract features on the driver's face. The system contains a section that controls the opening of a security gate at the exit. If the driver's face at the exit does not match the exit then the system will not allow the security gate to be opened. Based on the results of research that has been conducted on 372 license plate detection images and 169 images for character recognition, the success rate of license plate detection is 91.86% with an average IOU of 74.15% and character recognition reaching 95.8% with an average IOU of 81.25 %. The total system computing time is 4.85 seconds on Google Colabs and 37.69 seconds on Jetson Nano.

Keywords: *smart parking, artificial intelligence, YOLO, face recognition, barrier gate*

I. PENDAHULUAN

Manajemen sistem perparkiran pada umumnya masih menggunakan tenaga manusia untuk melakukan validasi menggunakan tiket karcis/STNK yang dicocokkan dengan plat nomor motor secara berulang-ulang. Hal ini dianggap masih tidak efisien karena dengan sistem tersebut terdapat celah keamanan dan proses kerja yang berulang. Tentunya dari sistem keamanan parkir dengan cara tersebut sudah sangat tidak efektif karena akan banyak memakan waktu dan *human error*. Hal ini diperkuat berdasarkan data statistik BPS, kasus pencurian kendaraan bermotor di Indonesia dari tahun 2015-2019 menyentuh angka rata-rata 32.538 kasus setiap tahunnya [1].

Diterima: 30 April 2022; Direvisi: 12 Oktober 2022; Disetujui: 8 November 2022

© 2022 JTERA (Jurnal Teknologi Rekayasa), Politeknik Sukabumi

Beberapa penelitian telah membahas sistem parkir yang terfokus pada wilayah area parkir dan posisi area parkir. Sementara pada bagian pintu masuk dan keluar area parkir sedikit yang membahas dari segi keamanan area tersebut. Pada bagian ini akan dipaparkan beberapa metode yang telah digunakan, khususnya pada bagian yang berhubungan dengan *Automatic License Plate Recognition* (ALPR) dan teknologi *face recognition*.

Beberapa peneliti telah membahas deteksi objek berupa plat nomor dengan menggunakan berbagai teknologi *Convolutional Neural Network* (CNN). Penelitian yang ditulis oleh [2] menggunakan teknologi *You Look Only Once* (YOLO) untuk melakukan deteksi plat nomor. Metode ini memiliki keunggulan pada kondisi ekstrim seperti pada cuaca buruk, minim pencahayaan, dan lalu lintas. Penelitian [3] menggunakan metode *Support Vector Machine* (SVM) dan *Region-based* CNN (RCNN) untuk deteksi plat dimana memiliki kelebihan untuk sistem yang membutuhkan *real time*.

Berbagai metode dinilai sudah mampu untuk melakukan pengenalan plat nomor salah satu yang sering digunakan adalah teknologi *Automatic Number Plate Recognition* (ANPR) yang mana menggunakan *Optical Character Recognition* (OCR) sebagai metode pengenalan karakter. Penelitian yang ditulis oleh Raed Abdulla dengan judul *Automatic Car Park Management System Using Face and Vehicle Registration Recognition*, dimana gambar pengendara dan plat nomor ditangkap oleh kamera dan diproses menggunakan matlab dengan pengenalan karakter menggunakan OCR. Pada penelitian ini didapatkan hasil pengamatan tidak lebih dari 3 detik dengan rentang akurasi 85-90%. Penelitian ini memiliki kekurangan pada gambar yang ter-*augmentasi* dimana sudut gambarnya tidak 0º dengan kamera dan hanya bekerja pada kondisi cahaya yang baik sehingga tidak sesuai dengan kenyataan di lapangan [4]. Peneliti [5] menggunakan OCR untuk pengenalan karakter. Dari total 25 gambar yang dites pada penelitian ini, didapatkan segmentasi karakter mencapai akurasi 90% (23 gambar dari 25 gambar) dan pengenalan karakter mencapai akurasi 96% (24 gambar dari 25 gambar). Penelitian ini pun memiliki kelemahan yang sama dimana belum mampunya sistem untuk mengenali karakter dengan sudut pengambilan gambar yang berbeda serta jumlah data tes yang sedikit. Penelitian yang dilakukan oleh [6] mendeteksi plat nomor yang mendeteksi kendaraan menggunakan teknik *deep learning* kemudian pada tahap akhir untuk meningkatkan pengenalan karakter yang kabur dan tidak jelas digunakan metode *LPR Convolution Neural Networks* (LPRCNN). Hasil eksperimen menunjukkan bahwa metodologi ini mencapai 96,12% tingkat deteksi kendaraan dan 94,23% deteksi plat. Menggunakan LPRCNN mencapai 99,2% karakter akurasi pengenalan.

Penelitian lainnya melakukan pengenalan terhadap kendaraan dan wajah menggunakan algoritma *Adaptive Boosting* dan *Haar-like features* sementara untuk identifikasi wajah menggunakan *Eigenfaces* untuk seleksi dan *Euclidian distance* untuk klasifikasi. Pada penelitian ini dibutuhkan perangkat yang cukup mahal untuk merealisasikannya sehingga dapat menghasilkan hasil komputasi yang cepat karena melewati berbagai proses yang kompleks [7]. Penelitian [8] menggunakan metode Viola-Jones untuk pendeteksi wajah. Hasil dari penelitian ini cukup akurat pada pose frontal dengan 3 macam kondisi cahaya dengan tingkat akurasi rata-rata mencapai 92,88%. Namun, tingkat akurasi rata-rata turun pada pose yang berbeda menjadi 85,99%. Metode Viola-Jones ini sangat sensitif terhadap perubahan sudut.

Terakhir, penelitian [9] menggunakan sistem ANPR pada kamera untuk mengelola, memantau, dan melindungi fasilitas parkir dalam memudahkan pengguna mencari kendaraannya. Pada penelitian ini hanya berfokus pada kemudahan untuk mencari lahan parkir yang kosong.

Berdasarkan karya-karya yang telah ada dapat dilihat bahwa penelitian sebelumnya terfokus pada teknologi pengenalan plat, pengenalan wajah, dan lokasi untuk lahan parkir. Sedikit penelitian yang fokus pada keamanan pada bagian gerbang keamanan. Padahal, proses identifikasi sangat penting pada bagian gerbang masuk karena disinilah lalu lintas kendaraan dapat diidentifikasi dengan baik untuk memperketat pengawasan kendaraan. Hal ini menjadi penting untuk tempat-tempat yang membutuhkan keamanan lebih, seperti gedung pemerintahan, tempat militer, area perbelanjaan, dan perumahan. Oleh karena itu, penelitian ini memiliki perbedaan metode dimana menggunakan arsitektur YOLOv4 sehingga memiliki kemampuan pengenalan objek secara *real time* karena memiliki komputasi yang cepat dan melakukan pencocokkan data dengan wajah sehingga terciptalah suatu sistem parkir automatis yang mampu membuka palang parkir keluar secara automatis apabila pengguna kendaraan memiliki kecocokkan data wajah dan plat yang sama dengan di pintu masuk.

II. METODE PENELITIAN

Pemilihan metode YOLOv4 didasari karena arsitektur YOLO memiliki kemampuan dalam segi akurasi, *augmentasi* gambar, dan dari segi

kecepatan YOLOv4 mampu mendeteksi *real time* dengan spesifikasi GPU yang mumpuni. Sehingga YOLOv4 dirasa sangat tepat untuk kasus ini. Pada bagian deteksi dan pengenalan wajah digunakan library face_recognition yang mana menggunakan teknologi *Histogram of Oriented Gradient* (HOG) [10] untuk mendeteksi lokasi wajah dan CNN untuk pengenalan wajah. Metode ini dinilai lebih andal untuk saat ini. Penggunaan library ini didasarkan pada kemampuannya dalam mengatasi gambar wajah dengan pose yang berbeda karena telah menggunakan algoritma yang disebut dengan *face landmark estimation* [11]. Alasan lainnya untuk menggunakan *library* ini adalah kebutuhan akan kemampuan mesin mendeteksi kemiripan wajah dengan hanya 1 gambar pada bagian input. Sehingga teknologi ini hanya membutuhkan waktu mili detik untuk melakukan deteksi wajah.

A. Tahapan Diagram Alur Penelitian

Pada Gambar 1 terdapat proses pada alur penelitian dimana output dari hasil penelitian ini berupa bobot yang terdiri dari 2 bobot yaitu untuk deteksi dan untuk karakter plat nomor, yang mana pada akhir dari penelitian ini merupakan integrasi model dengan aplikasi dan *database* dengan *library* face_recognition.

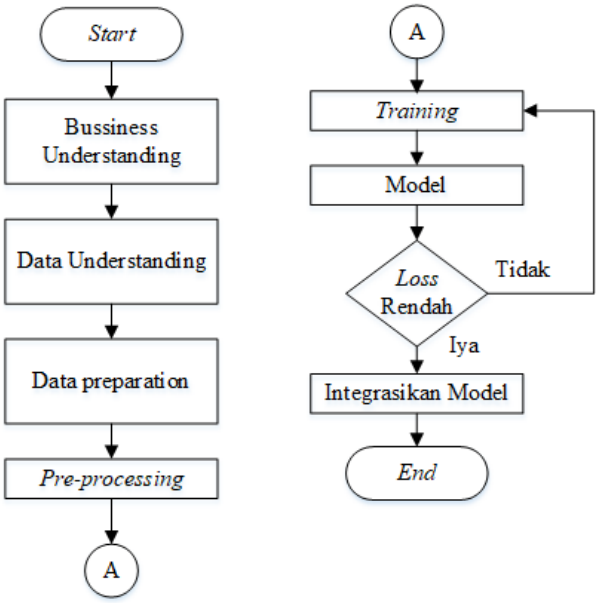
Pada tahap *bussiness understanding* merupakan tahapan untuk memahami permasalahan yang ada, dimana menetapkan tujuan dari pengumpulan data yang diperlukan. Permasalahan yang ada merupakan permasalahan keamanan tambahan pada area gerbang masuk menuju area parkir yang mana permasalahan ini ditujukan untuk area sensitif yang memerlukan keamanan tambahan. Sehingga terciptalah ide untuk menuntaskan permasalahan ini dengan menerapkan sistem gerbang parkir dengan pencacatan otomatis memanfaatkan data plat nomor yang dicocokan dengan wajah dimana dapat diselesaikan dengan arsitektur YOLOv4 dan face_recognition.

Tahap *data understanding* peneliti mengumpulkan data yang diperlukan berupa gambar kendaraan bermotor dan gambar plat yang telah dilakukan *pre-processing*. *Dataset* gambar kendaraan bermotor berjumlah 2160 gambar dan *dataset* gambar plat nomor berjumlah 1049 gambar. Data gambar ini diperoleh dari observasi pribadi dan berasal dari *Google's Open Image Dataset*. Tabel 1 merupakan tabel data pembagian dataset.

Data preparation merupakan proses konversi format gambar menjadi JPG, proses pemfilteran gambar dan proses pelabelan sebelum dilakukan *training* dapat dilihat pada Gambar 2 dan Gambar 3. Khusus untuk *dataset* plat nomor dilakukan proses

Tabel 1. Pembagian *dataset*

No.	Nama *dataset*	Jenis data	Jumlah data	Format data
1	*Dataset* deteksi plat	Data *training*	1788	JPG
2	*Dataset* deteksi plat	Data tes	372	JPG
3	*Dataset* pengenalan plat	Data *training*	880	JPG
4	*Dataset* pengenalan plat	Data tes	169	JPG



Gambar 1. Tahapan diagram alur penelitian

pre-processing pada Gambar 4 sebelum dilakukan proses *training*. Proses *pre-processing* sendiri terdiri dari koreksi warna gambar (*shadow*, *highlight*, *brightness*), mengubah gambar menjadi *gray*, menghaluskan gambar dengan *blur* dan *denoise*, koreksi *gamma* dan *histogram*, dan *threshold*. Hal itu bertujuan agar memudahkan mesin dalam mencari pola yang terdapat pada gambar.

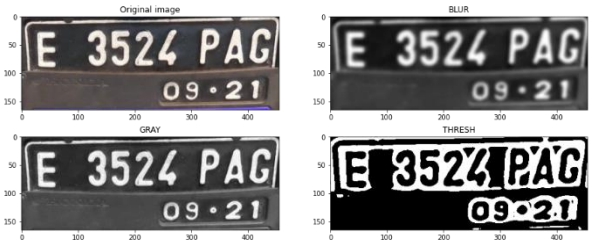
Setelah semua proses persiapan dilakukan, langkah selanjutnya adalah melakukan proses *training*, dimana proses *training* ini menggunakan *framework* darknet yang dibuat oleh AlexeyAB yang untuk kemudian kita simpan ke dalam Google Drive. Proses *training* ini membutuhkan waktu komputasi yang lama sehingga menggunakan Google Colaboratory yang mana memiliki GPU yang *powerful* sehingga memungkinkan proses *training* lebih cepat dan minim terjadi *error*. Pada tahapan ini pelatihan akan terus diulang sampai mendapatkan model yang memiliki nilai *Mean Average Precission* (mAP) atau *average loss* rendah.



Gambar 2. Proses pelabelan *dataset* deteksi plat



Gambar 3. Proses pelabelan *dataset* pengenalan plat

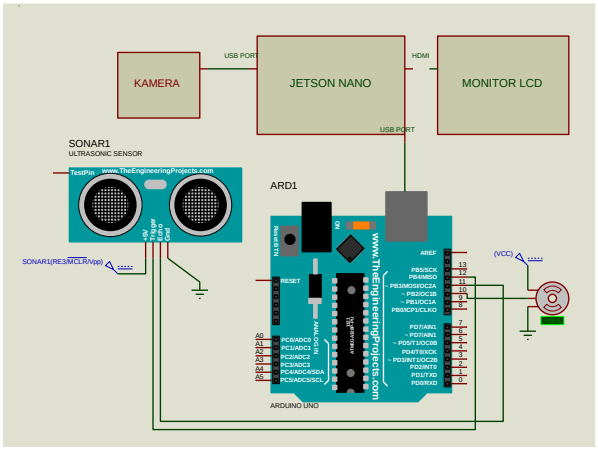


Gambar 4. Tahapan *pre-processing* plat nomor

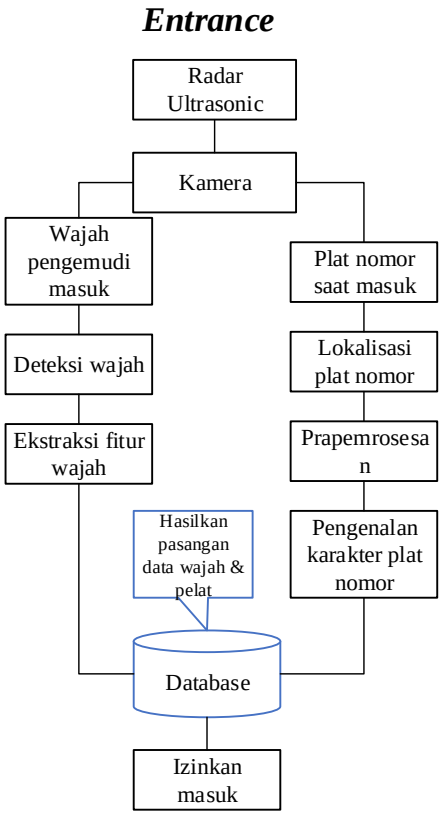
B. Blok Diagram dan Skema Rangkaian

Pada Gambar 5 dapat kita lihat bahwa jetson nano digunakan sebagai *processor* dan Arduino digunakan sebagai *microcontroller* yang terhubung dengan servo dan sensor ultrasonik. Sensor ultrasonik sendiri digunakan untuk input apakah di depan gerbang yang disimulasikan dengan servo ini terdapat kendaraan atau tidak. Apabila terdapat kendaraan maka kamera akan melakukan *capture* gambar untuk kemudian diproses menghasilkan data ekstraksi wajah dan plat nomor. Sebetulnya pada Jetson Nano terdapat pin yang dapat mengolah data sensor ultrasonik, namun itu tidak bias digunakan karena Jetson Nano akan menerima input masukan berupa tegangan yang dapat membuat jetson nano *short circuit*. Oleh karena itu, digunakanlah arduino yang nantinya akan dihubungkan ke jetson nano menggunakan USB tipe B yang nantinya komunikasi jetson nano dengan arduino adalah dengan menggunakan *library* PyFirmata untuk membaca *input* dan *output* dari sensor ultrasonik.

Pada Gambar 6 hal yang pertama dilakukan oleh sistem adalah mendeteksi apakah ada kendaraan atau tidak. Hal itu dibuat agar sistem tidak *overheat* karena proses *image processing* yang terus menerus



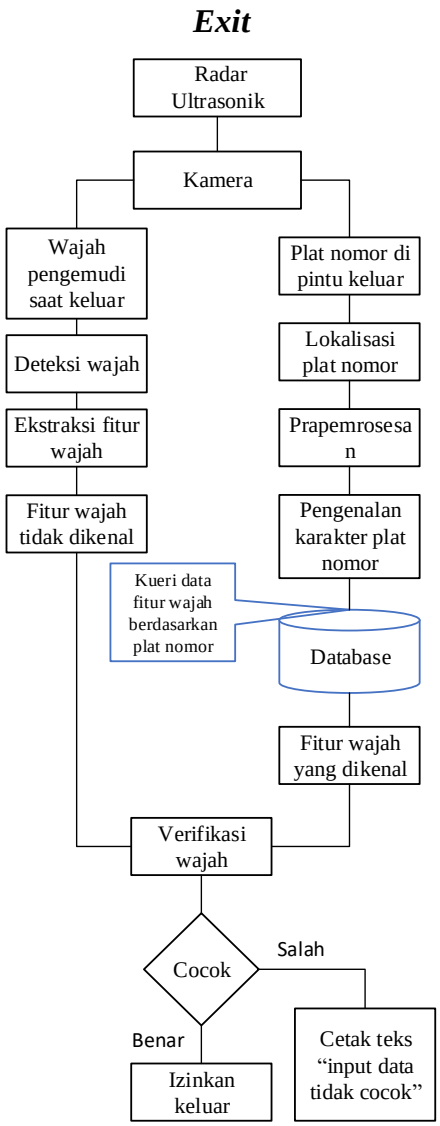
Gambar 5. Blok diagram skema elektronik



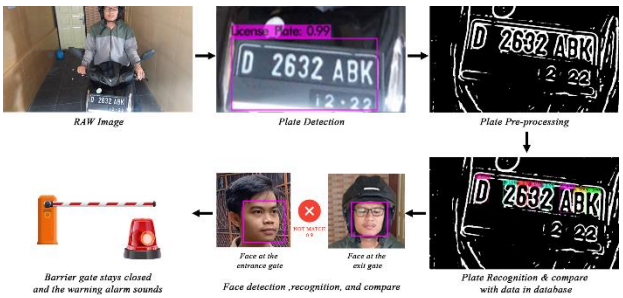
Gambar 6. Blok diagram pada gerbang masuk

dioperasikan di dalam GPU. Sehingga dengan adanya deteksi jarak terlebih dahulu, ini menghemat beban GPU berkali-kali lipat. Setelah itu kamera akan melakukan *capture* gambar. Gambar yang diperoleh selanjutnya akan melewati proses pengenalan karakter dan juga pengenalan wajah. Setelah didapatkan data wajah dan karakter selanjutnya akan disimpan ke dalam *database* dan gerbang terbuka.

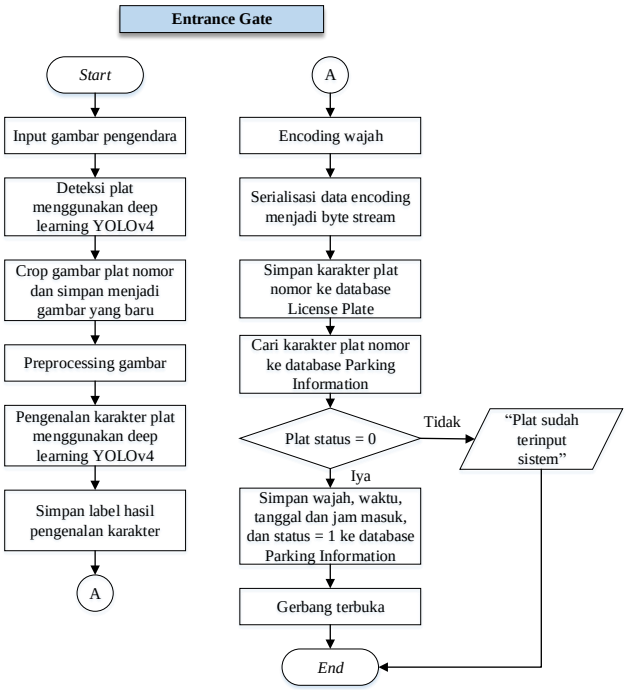
Gambar 7 menunjukkan proses ketika akan keluar, dimana proses yang sama dilakukan dengan melakukan pengenalan wajah dan pengenalan karakter plat nomor. Hanya saja untuk membuka gerbang keluar, data wajah harus sama dengan data pada saat masuk.



Gambar 7. Blok diagram pada gerbang keluar



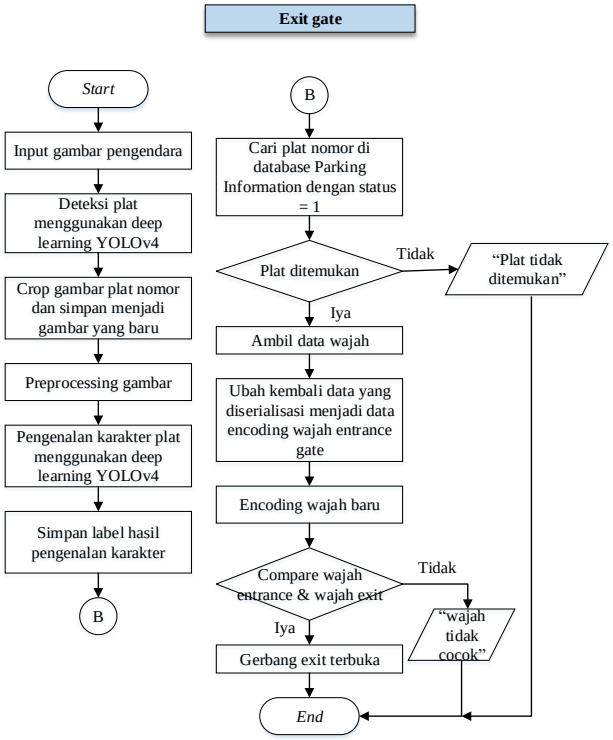
Gambar 8. Alur sistem bekerja



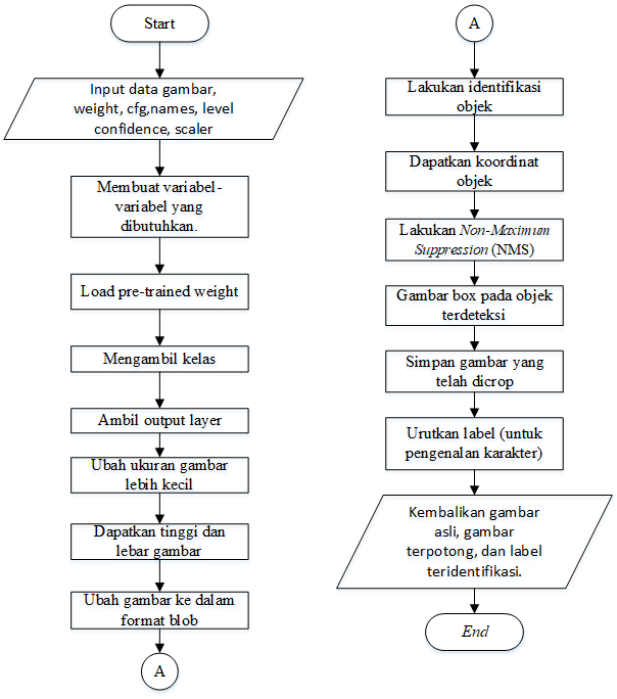
Gambar 9. Algoritma pada gerbang masuk

C. Algoritma

Gambar 8 dan Gambar 9 menunjukkan algoritma dimana pertama-tama *input* gambar pengendara akan dideteksi menggunakan algoritma *deep learning* YOLOv4 untuk selanjutnya dilakukan deteksi objek plat nomor pada gambar. Apabila gambar terdeteksi maka sistem akan melakukan *cropping* gambar menjadi gambar yang baru untuk memudahkan proses pengenalan karakter plat nomor. Selanjutnya adalah *pre-processing* gambar dimana tujuannya untuk menghasilkan pola karakter yang lebih jelas sehingga mampu dikenali dengan baik oleh mesin. Pada bagian ini kita akan untuk memisahkan *background* dengan objeknya. Langkah ini dinamakan *thresholding* dimana kita akan melakukan konversi *grayscale image* ke *binary image* dimana pixelnya akan bernilai antara 0 atau 255. Pada proses ini nantinya akan ada nilai batas ambang, dimana jika kurang dari nilai batas, maka akan dikonversi menjadi 0 dan jika lebih dari nilai tersebut maka akan dikonversi menjadi 255 [12]. Proses selanjutnya merupakan pengenalan gambar menggunakan algoritma yang sama yaitu YOLOv4. Setelah gambar berhasil dikenali label pengenalan plat akan disimpan ke dalam suatu variabel sebagai hasil pengenalan karakter. Proses selanjutnya merupakan pengenalan wajah ini menggunakan *library* face_recognition dimana prinsip dasarnya merupakan *Histogram of Oriented Gradients*. Setelah didapatkan *array encoding* dari suatu wajah, *array* tersebut dikonversi menjadi *byte stream*. Tujuannya adalah agar *array* dapat disimpan ke dalam *database* tanpa mengubah isinya menjadi tipe data string. Selanjutnya simpan data karakter plat nomor ke dalam data *License Plate*. Untuk mencegah terjadinya *double record*, data harus dipastikan tidak ada yang tersimpan pada tabel ParkingInformation setelah itu barulah data dapat disimpan di dalam tabel berikut *encoding* wajah, nomor plat, waktu masuk, dan status. Status merupakan penanda bahwa kendaraan tersebut



Gambar 10. Algoritma pada gerbang keluar



Gambar 11. Pipeline YOLO

terparkir atau tidak. Apabila status = 1 maka kendaraan terparkir, dan apabila status = 0 maka kendaraan telah keluar. Setelah seluruh proses dilewati gerbang parkirpun terbuka.

Pada gerbang keluar proses (Gambar 10) yang sama dilakukan, perbedaannya adalah pada gerbang keluar terdapat verifikasi wajah. Apabila data wajah pada saat keluar *match* dengan data wajah pada saat masuk, maka gerbang pintu keluar parkir akan terbuka apabila tidak cocok maka akan sistem akan memberitahu bahwa wajah tidak cocok. Setelah proses training selesai, langkah selanjutnya adalah membuat YOLO *pipeline* (Gambar 11) agar nantinya program ini dapat fleksibel digunakan untuk pengenalan karakter dan deteksi plat.

Tabel 2. Spesifikasi perangkat

Parameter	Jetson Nano *Developer Kit*	*Google Colabs*
GPU	128-core Maxwell™ GPU	Nvidia K80 / T4 12 GB / 16GB 0,82 GHz / 1,59 GHz
CPU	Quad-core ARM® Cortex®-A57 CPU	Intel(R) Xeon(R) 2,30 GHz 2 Cores
Memori	4 GB 64-bit LPDDR4	12 GB (*upgradable* to 26,75GB)

III. HASIL DAN PEMBAHASAN

Sistem yang telah dibuat dan dilakukan pengujian dengan menggunakan jetson nano dan Google Colabs sebagai pembanding kecepatan dari masing-masing GPU dengan spesifikasi pada Tabel 2.

Data yang diuji merupakan data gambar dari kamera dengan parameternya seberapa baik sistem yang dapat melakukan deteksi plat nomor, pengenalan karakter pada plat nomor, dan pengenalan wajah pengendara kendaraan motor serta berapa lama waktu yang dibutuhkan dalam pemrosesannya dengan spesifikasi prosesor tertentu baik di gerbang masuk maupun di gerbang keluar parkir. Pengujian akan dilakukan berupa simulasi kendaraan motor yang masuk dan keluar melewati sistem gerbang ini.

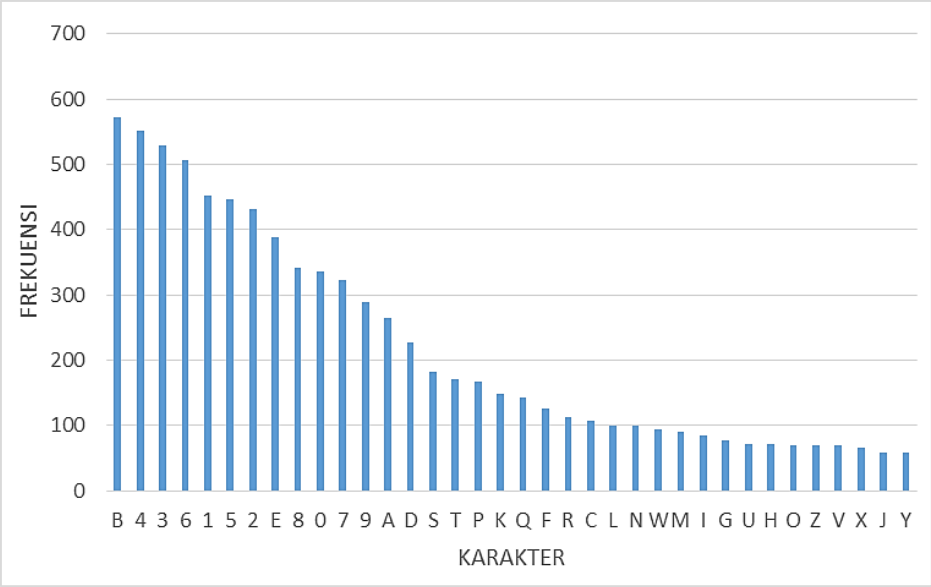
A. *Mean Average Precision* dan *Intersection Over Union*

Mean Average Precision (mAP) digunakan untuk mengukur performa dari kinerja model yang melakukan tugas untuk deteksi objek. Untuk deteksi objek kita menggunakan *Intersection Over Union* dimana kita mengukur tumpang tindihnya dua kotak dimana kotak yang pertama merupakan kotak yang benar, dan kotak yang kedua merupakan kotak prediksi [13]. Dari ke dua kotak yang saling tumpang tindih kita bisa mengukur berapa banyak kotak prediksi tumpang tindih dengan kotak kebenaran. Kita dapat mengatur nilai *threshold* untuk IoU untuk menentukan apakah deteksi objek itu benar atau tidak. Sehingga kita dapat menentukan nilai minimum dari IoU untuk dianggap prediksi itu match atau tidak. Jadi kita

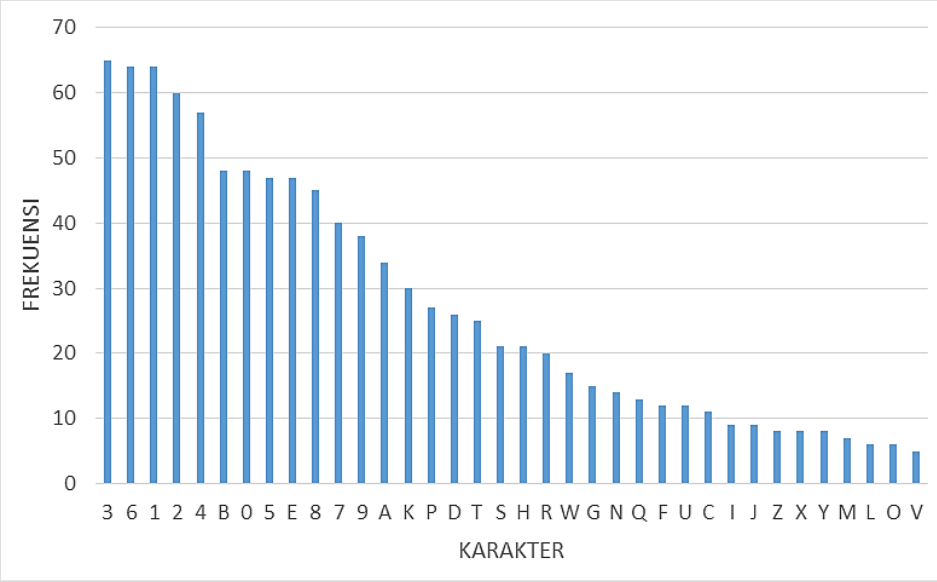
harus memberikan nilai untuk IoU untuk memutuskan apakah deteksi objek dalam keadaan valid atau tidak. Pada penelitian ini IoU diatur 0.5. Terdapat empat kemungkinan hasil dari pengujian yang didapat:

1. Apabila IoU >= 0,5, maka klasifikasi dan deteksi objek adalah *True Positive* (TP), prediksi dimana terdapat plat nomor atau karakter plat nomor.
2. Apabila tidak ada objek dalam gambar dan model tidak memprediksi objek apapun, maka disebut dengan *True Negative* (TN), prediksi ini contohnya adalah tidak terdapat plat nomor atau karakter plat nomor.
3. Apabila IoU < 0,5, maka klasifikasi dan deteksi objek adalah *False Positive* (FP), contohnya kondisi dimana tidak ada plat nomor namun terdeteksi terdapat plat nomor atau karakter plat nomor.
4. Apabila sebenarnya dalam gambar terdapat objek dan model gagal mendeteksi objek, maka kita klasifikasikan *False Negative* (FN), contohnya adalah kondisi dimana terdapat plat atau karakter plat nomor namun tidak terdeteksi.

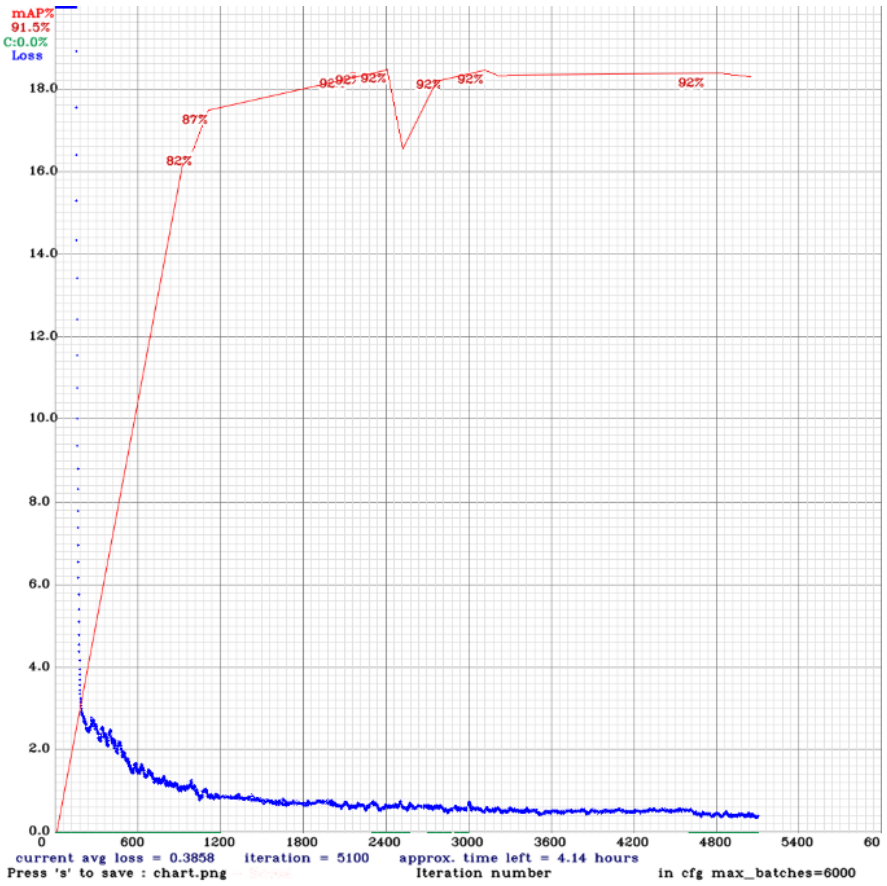
Gambar 12 dan Gambar 13 merupakan distribusi jumlah karakter yang dilakukan proses *training* dan evaluasi. Gambar 14 dan Gambar 15 merupakan gambar yang menunjukan mAP dan *average loss* untuk dataset deteksi plat nomor dan pengenalan karakter plat nomor.



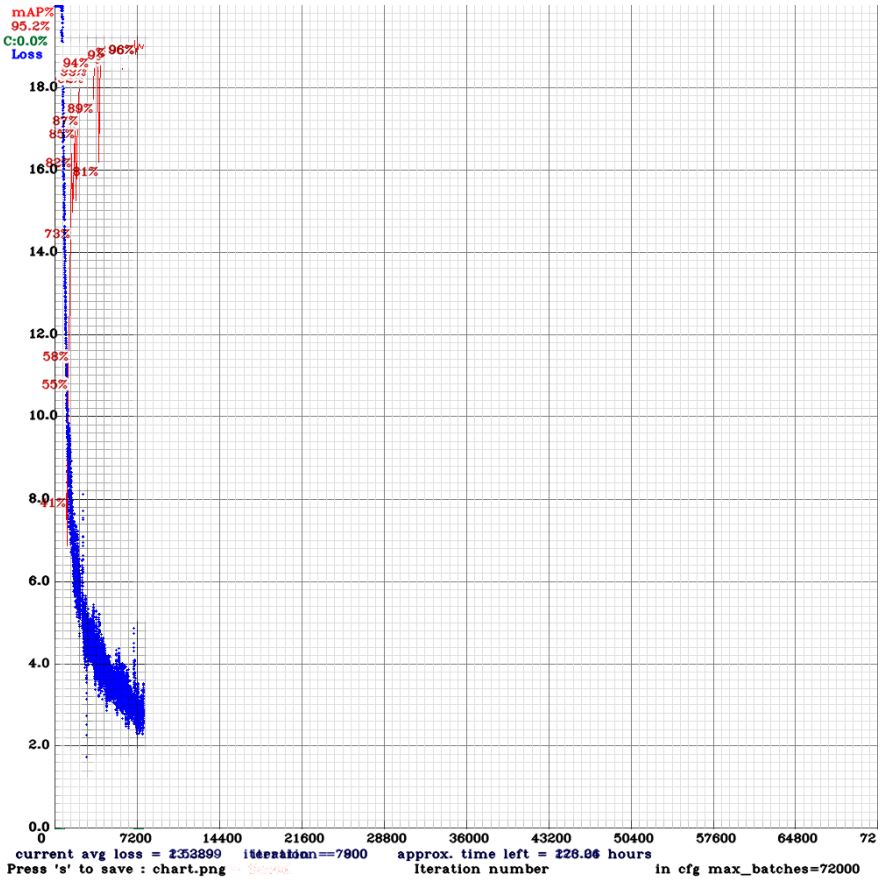
Gambar 12. Persebaran data karakter *training*



Gambar 13. Persebaran data karakter evaluasi



Gambar 14. Grafik mAP dan *average loss* deteksi plat



Gambar 15. Grafik mAP dan *average loss* pengenalan plat

B. Confusion Matrix

Evaluasi performa dari metode klasifikasi menggunakan parameter *Confusion Matrix* [14]:

1. *Accuracy* adalah rasio subjek yang diberi label dengan benar ke seluruh kumpulan subjek.
2. *Precision* adalah rasio dari subjek benar yang diberi label dengan benar oleh mesin dengan semua subjek benar yang diberi label.
3. *Recall* atau biasa disebut juga dengan sensitifitas adalah rasio dari gambar plat nomor yang diberi label dengan benar oleh mesin dengan semua gambar yang ada plat nomornya baik yang terdeteksi maupun yang tidak terdeteksi
4. *F1-score* mempertimbangkan *precision* dan *recall*. Ini adalah rata-rata harmonik (rata-rata) dari *precision* dan *recall*

Berdasarkan Tabel 3 dan Tabel 4 didapatkan deteksi plat nomor terbaik adalah pada bobot 3.000 dimana memiliki mAP tertinggi 92,22%. Kemudian pada Tabel 5 dan Tabel 6 didapatkan pengenalan karakter terbaik adalah pada bobot 6.500 dimana memiliki mAP tertinggi 95,83%.

Tabel 3. *Mean average precision* dan *average intersection over union* untuk deteksi plat

Weight	mAP (%)	*Prec*	*Recall*	F1-Score	Av IOU (%)
1000	82,44	0,74	0,85	0,79	56,41
2000	92,03	0,9	0,89	0,9	72,73
3000	92,22	0,9	0,9	0,9	74,15
4000	91,76	0,92	0,9	0,91	75,2
5000	91,7	0,92	0,89	0,9	76,95
Best	91,86	0,9	0,9	0,9	74,15
Last	91,58	0,91	0,89	0,9	76,55

Tabel 4. *True Positive, False Positive*, dan *False Negatif* untuk deteksi plat

Weight	TP	FP	FN
1000	504	48	62
2000	507	58	59
3000	502	46	64
4000	509	47	57
5000	508	56	58
Best	504	55	62
Last	480	172	86

Tabel 5. *Mean average precision* dan *average intersection over union* untuk pengenalan plat

Weight	mAP (%)	*Prec*	*Recall*	F1-Score	Av IOU (%)
2500	93.08	0,85	0,95	0,89	70,43
3000	94,01	0,90	0,95	0,92	76,88
3500	94,12	0,90	0,95	0,92	74,59
4000	94,40	0,89	0,95	0,92	77,03
4500	95,27	0,91	0,95	0,93	78,58
5000	94,76	0,92	0,95	0,94	80,19
6000	95,22	0,92	0,96	0,94	79,18
6500	95,83	0,92	0,96	0,94	81,25
Best	95,63	0,92	0,96	0,94	75,20

Tabel 6. *True Positive, False Positive*, dan *False Negatif* untuk pengenalan plat

KELAS	AP (%)	TP	FP
A	94,44	62	9
B	98,82	77	11
C	94,79	15	2
D	97,95	48	8
E	93,59	52	1
F	94,77	23	1
G	94,54	19	1
H	100,00	29	7
I	81,99	16	1
J	100,00	9	0
K	99,08	47	9
L	93,08	16	3
M	94,05	11	0
N	95,14	23	12
O	82,94	6	1
P	82,94	6	1
Q	91,62	20	12
R	91,91	23	0
S	100,00	25	2
T	96,00	31	0
U	98,36	15	3
V	91,43	8	2
W	95,49	29	6
X	85,34	8	0
Y	90,00	9	2
Z	98,06	13	1
1	99,78	100	4
2	100,00	83	0
3	99,94	90	4
4	99,83	107	5
5	100,00	75	2
6	98,31	83	6
7	99,97	55	1
8	99,55	69	5
9	93,50	54	9
0	98,56	76	0

Tabel 7. Kecepatan komputasi pada Google Colabs dan Jetson Nano

Komputer	Deteksi Plat nomor (ms)	Pre-processing	Pengenalan karakter Plat nomor (ms)	Pengenalan Wajah (ms)	Sistem (ms)
Colabs	1,649924	0,145652	1,63441	1,63441	4,859
Jetson Nano	10,88733	0,341682	10,6025	15,45032	37,69



Gambar 16. GUI pada gerbang masuk (kiri) dan gerbang keluar (kanan)

C. Pengujian Waktu Komputasi dan GUI

Pengujian dilakukan pada Google Colabs dengan Jetson Nano untuk mendapatkan perbandingan hasil menggunakan 2 GPU yang berbeda. Hasil pengujian dapat dilihat pada Tabel 7. Berdasarkan hasil pengujian diperoleh bahwa Google Colabs menghasilkan kemampuan komputasi yang lebih cepat jika dibandingkan dengan Jetson Nano.

Gambar 16 merupakan tampilan GUI pada gerbang masuk dan gerbang keluar. Pada GUI terdapat proses *pre-processing* plat nomor sebelum dibaca oleh YOLOv4. Gambar wajah kemudian ditampilkan di GUI untuk dilihat oleh *use*r apakah wajah yang diambil sudah benar atau belum.

IV. KESIMPULAN

Sistem keamanan parkir automatis berhasil dibuat dengan memiliki kemampuan deteksi plat dengan karakter pada berbagai kondisi ekstrim seperti sudut pengambilan gambar dan cahaya. Juga memiliki kemampuan terhadap gambar wajah pada berbagai sudut gambar sehingga sistem dapat bekerja maksimal dalam melakukan proses pengenalan wajah. Sistem ini memiliki tingkat mAP 91,86% dengan average IOU 74,15% untuk deteksi plat dan mAP 95,83% dengan average IOU mencapai 81,25% untuk pengenalan karakter. Hasil ini sudah diuji pada 372 gambar uji deteksi plat nomor dan 169 gambar uji pengenalan karakter. Dari pengenalan wajah kemampuan sistem dalam melakukan pengenalan wajah pada 29 data uji dengan baik. Selain itu sistem memiliki kelebihan dalam kecepatan komputasi. Pada Google Colabs rata-rata waktu yang dibutuhkan oleh sistem adalah 4,8 detik, sedangkan pada Jetson Nano membutuhkan 37,69 detik. Penelitian yang telah dibuat ini dapat dikembangkan penerapannya pada cakupan area yang lebih luas seperti pada area lalu lintas.

REFERENSI

[1] Badan Pusat Statistik, "Statistik Kriminal 2020," *Badan Pusat Statistik*, Jakarta, 2020.

[2] G. S. Hsu, A. Ambikapathi, S. L. Chung, and C. P. Su, "Robust license plate detection in the wild," *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-6, 2017.

[3] M. A. Rafique, W. Pedrycz, and M. Jeon, "Vehicle license plate detection using region-based convolutional neural networks," *Soft Computing*, vol. 22, no. 19, pp. 6429-6440, 2018.

[4] G. Subhashini, M. E. Markhi, and R. Abdulla, "Automatic Car Park Management System Using Face and Vehicle Registration Recognition," in *Proceedings of The Fourth International Conference on Technological Advances in Electrical, Electronics and Computer Engineering*, Malaysia, 2016.

[5] K. Jain, T. Choudhury, and N. Kashyap, "Smart vehicle identification system using OCR," in *2017 3rd international conference on computational intelligence & communication technology (CICT)*, Ghaziabad, 2017.

[6] C. H. Lin, Y. S. Lin, and W. C. Liu, "An efficient license plate recognition system using convolution neural networks," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, Chiba, pp. 224-227, 2018.

[7] Z. Mahmood, T. Ali, S. Khattak, S. U. Khan, and L. T. Yang, "Automatic Vehicle Detection and Driver Identification Framework for Secure Vehicle Parking," in *2015 13th International Conference on Frontiers of Information Technology (FIT)*, Islamabad, 2015.

[8] I. Taleb, M. E. A. Ouis, and M. O. Mammar, "Access control using automated face recognition: Based on the PCA & LDA algorithms," in *2014 4th International Symposium ISKO-Maghreb: Concepts and Tools for knowledge Management (ISKO-Maghreb)*, Algiers, pp. 1-5, 2015.

[9] M. Y. Aalsalem, W. . Z. Khan, and K. M. Dhabbah, "An automated vehicle parking monitoring and management system using ANPR cameras," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, Pyeong Chang, pp. 706-710, 2015.

[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Lecture Notes in Computer Science*, pp. 428-441, 2006.

[11] A. Geitgey, "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning," Medium, 24 July 2016. [Online]. Available: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78.

[12] A. Rosebrock, "OpenCV Thresholding (cv2.threshold)," pyimagesearch, 28 April 2021. [Online]. Available: https://pyimagesearch.com/2021/04/28/opencv-thresholding-cv2-threshold/.

[13] S. Yohanandan, "mAP (mean Average Precision) might confuse you!," Towards Data Science, 9 June 2020. [Online]. Available: https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2.

[14] S. Narkhede, "Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?," Towards Data Science, 2 April 2019. [Online]. Available: https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124.

[15] A. K. Jain and S. Z. Li, "Handbook of face recognition," Springer, vol. I, 2011.