DETEKSI KETERSEDIAAN TEMPAT PARKIR MENGGUNAKAN MASK R-CNN (STUDI KASUS: FAKULTAS TEKNOLOGI INDUSTRI UNISSULA)

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang

DISUSUN OLEH :
AHMAD AYUDHAWARA
NIM 32601800002

FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2024

FINAL PROJECT
PARKING DETECTION SYSTEM USING MASK R-CNN (CASE STUDY : FACULTY OF TECHNOLOGY INDUSTRY UNISSULA)

Proposed to complete the requirement to obtain a bachelor's degree (S1)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University

ARRANGED BY :

AHMAD AYUDHAWARA

NIM 32601800002

INDUSTRIAL TECHNOLOGY FACULTY

SULTAN AGUNG ISLAMIC UNIVERSITY

SEMARANG

2024

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **"Deteksi Ketersediaan Tempat Parkir Menggunakan Mask-RCNN (Studi Kasus Fakultas Teknologi Industri UNISSULA)"** ini disusun oleh :

Nama : Ahmad Ayudhawara

NIM : 32601800002

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari : jumat

Tanggal : 7 juni 2024

Mengesahkan,

Pembimbing I	Pembimbing II
Bagus Satrio WP, S.Kom., M.Cs	Muhammmad Qomaruddin, ST.MSc, P.hD
NIDN. 1027118801	NIDN. 0631057101

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung

Ir. Sri Mulyono, M.Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul **"Deteksi Ketersediaan Tempat Parkir Menggunakan Mask R-CNN (Studi Kasus : Fakultas Teknologi Industri UNISSULA)"** ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : jumat

Tanggal : 7 juni 2024

TIM PENGUJI

Anggota I	Anggota II
Ir.Sri Mulyono, M.Eng	Moch Taufik, ST.MIT
NIDN. 0626066601	NIDN. 0622037502

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama	: Ahmad Ayudhawara
NIM	: 32601800002
Judul Tugas Akhir	: Deteksi Ketersediaan Tempat Parkir Menggunakan Mask R-CNN (Studi Kasus : Fakultas Teknologi Industri UNISSULA)

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apbila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 5 Juni 2024

Yang Menyatakan,

Ahmad Ayudhawara

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Ahmad Ayudhawara

NIM : 32601800002

Program : Teknik Informatika

Studi Fakultas : Teknologi industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Deteksi Ketersediaan Tempat Parkir Menggunakan Mask R-CNN (Studi Kasus : Fakultas Teknologi Industri UNISSULA)

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 5 Juni 2024

Yang menyatakan,

Ahmad Ayudhawara

KATA PENGANTAR

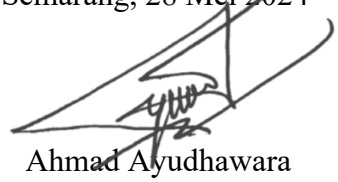
Dengan mengucap syukur alhamdulillah atas kehadirat Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul "Deteksi Ketersediaan Tempat Parkir Menggunakan Mask-RCNN (Studi Kasus : Fakultas Teknologi Industri UNISSULA)" ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, S.T., M.T.
3. Dosen pembimbing I penulis Bagus Satrio WP, S.Kom., M.Cs yang telah meluangkan waktu dan memberi ilmu.
4. Dosen pembimbing II penulis Muhammmad Qomaruddin, ST.MSc, P.hD yang memberikan banyak nasehat dan saran.
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini,
6. Dan kepada semua pihak yang tidak dapat saya sebutkan satu persatu.

Dengan rendah hati, penulis menyadari bahwa laporan masih memiliki banyak kekurangan dalam hal kuantitas, kualitas, dan ilmu pengetahuan. Oleh karena itu, penulis mengharapkan kritikan dan saran yang membangun untuk membantu laporan ini menjadi lebih baik di masa depan.

Semarang, 28 Mei 2024



Ahmad Ayudhawara

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Tempat parkir merupakan sarana yang dibutuhkan hampir oleh setiap orang, dan lagi jumlah pemilik kendaraan bermotor di Indonesia terus meningkat setiap tahunnya. Hal tersebut dapat menjadi suatu permasalahan untuk pengguna kendaraan untuk dapat menemukan lahan parkir yang semakin sedikit. Di zaman sekarang ini sudah banyak metode yang digunakan untuk dapat memudahkan sistem perparkiran, salah satunya dengan menggunakan ilmu citra digital. Sistem ini menggunakan metode deteksi objek pada citra digital yaitu Mask R-CNN dimana metode tersebut dapat mendeteksi banyak objek pada satu frame dengan hasil keluaran yang lebih lengkap. Sistem ini dibangun agar memudahkan pengguna kendaraan mobil untuk mencari lahan parkir yang masih kosong. Pengujian sistem ini menggunakan empat sampel video dengan tempat dan kondisi yang berbeda. Hasil pengujian menunjukkan untuk rata-rata akurasi ketersediaan tempat parkir sebesar 94,3%, namun hasil evaluasi untuk deteksi objek mobil masih belum bagus serta rata-rata FPS yang didapatkan hanya sejumlah 0,36 FPS.

Kata kunci : mask r-cnn, parkir, deteksi objek.

ABSTRACT

Parking is a facility that almost everyone needs, and the number of motor vehicle owners in Indonesia continues to increase every year. This can be a problem for vehicle users to find fewer and fewer parking spaces. In this day and age, many methods are used to facilitate the parking system, one of which is using digital image science. This system uses an object detection method in digital images, namely Mask R-CNN, where this method can detect many objects in one frame with more complete output results. This system was built to make it easier for car users to find empty parking lots. Testing this system uses four video samples with different places and conditions. The test results show that the average accuracy of parking space availability is 94.3%, however the evaluation results for car object detection are still not good and the average FPS obtained is only 0.36 FPS.

Keywords: mask r-cnn, parking, object detection.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Jumlah kendaraan bermotor di Indonesia terus meningkat setiap tahunnya, baik dari kendaraan bermotor seperti mobil penumpang, sepeda motor, mobil bus, dan mobil barang. Sesuai data jumlah kendaraan bermotor milik BPS (Badan Pusat Statistik), jumlah kendaraan di Indonesia pada tahun 2020 mencapai 136.137.451. Dihimpun dari data tersebut pada tahun 2018 jumlah kendaraan mobil penumpang mencapai 14.830.698, tahun 2019 sebanyak 15.592.419, dan pada tahun 2020 mencapai 15.797.746 mobil. Disini dapat dilihat bahwa mobil penumpang menyumbang 11,6 persen dari total jumlah kendaran di Indonesia (Statistik, 2020).

Seiring dengan bertambahnya jumlah kendaraan bermotor di Indonesia tentunya lahan parkir yang kosong juga semakin dibutuhkan oleh para pengguna kendaran tersebut. Hal ini mengakibatkan banyaknya kendaraan bermotor yang memarkirkan kendaraannya tidak sesuai pada tempatnya. Masalah tersebut juga sering terjadi akibat tidak adanya informasi mengenai penuh atau tidaknya lahan parkir yang tersedia di suatu tempat, seperti pusat perbelanjaan, kantor, tempat wisata, dan lain-lain. Akibat tidak adanya informasi mengenai ketersediaan lahan parkir pada tempat-tempat tersebut, seringkali pengendara bermotor harus berkeliling terlebih dahulu untuk mencari tempat parkir yang masih kosong. Namun jika tidak ditemukan tempat parkir yang tersedia, pengendara terpaksa untuk mencari tempat parkir di tempat yang lain atau bahkan memaksakan untuk parkir sembarangan di tempat yang tidak tepat.

Di zaman sekarang ini kemajuan teknologi terutama dalam lingkup pengolahan citra sudah sangat modern. Pengolahan citra adalah suatu teknik yang digunakan untuk memproses dan memanipulasi sebuah citra digital untuk mendapatkan informasi tertentu dari citra yang diproses (Yusnita dkk, 2012). Salah satu algoritma untuk mendeteksi suatu objek pada citra adalah *Mask* R-CNN. *Mask* R-CNN adalah *Convolutional Neural Network* (CNN) dan

merupakan bagian dalam hal segmentasi gambar serta *instance segmentation*. *Mask* R-CNN merupakan perkembangan dari *Faster* R-CNN. Jika dibandingkan dengan *Faster* R- CNN, *Mask* R-CNN memberikan tiga keluaran untuk setiap objek kandidat, yaitu label kelas, objek pembatas, dan objek topeng (*mask*). Hal tersebut tentunya membuat algoritma deteksi objek dari *Mask* R-CNN lebih baik karena tedapat variabel tambahan yaitu objek topeng yang tentunya membuat deteksi dari objek yang diinginkan lebih baik. *Mask* R-CNN juga memiliki beberapa kelebihan diantaranya mudah untuk dilatih, mengungguli semua entri model tunggal yang ada pada setiap tugas, sangat efisien dan hanya menambahkan sedikit tugas diatas *Faster* R-CNN, dan yang terakhir *Mask* R-CNN mudah di generalisasi ke tugas yang lain. Salah satu manfaat dari algoritma ini adalah untuk mendeteksi ketersediaan slot parkir dengan cara mengambil informasi berupa deteksi citra mobil pada lahan parkir yang ada. Sayangnya saat ini sistem parkir yang ada pada kampus Universitas Islam Sultan Agung belum menggunakan sistem digital dimana prosesnya hanya memberikan karcis untuk pengguna kendaraan bermotor dan tidak ada sistem untuk memberikan informasi mengenai ketersediaan tempat parkir di wilayah tersebut.

Dari pemaparan masalah tersebut dibutuhkan solusi untuk memberikan informasi kepada pengguna kendaraan bermotor mengenai ketersediaan lahan parkir di suatu tempat, oleh karena itu penulis mengusulkan Tugas Akhir dengan judul Sistem Deteksi Ketersediaan Tempat Parkir menggunakan *Mask* R-CNN (Studi Kasus : Fakultas Teknologi Industri Universitas Islam Sultan Agung). Sistem ini akan digunakan untuk mendeteksi ketersediaan lahan parkir yang masih kosong di wilayah FTI UNISSULA yang diharapkan dapat membantu para pengendara mobil agar lebih mudah untuk mengetahui ketersediaan tempat parkir di wilayah tersebut, serta dapat membantu memanajemen pengelolaan parkir agar lebih efektif.

1.2 Rumusan Masalah

Bagaimana algoritma *Mask* R-CNN dapat diterapkan untuk mendeteksi ketersediaaan tempat parkir pada Fakultas Teknologi Industri Universitas Islam Sultan Agung.

1.3 Batasan Masalah

Berdasarkan rumusan yang telah diuraikan, batasan masalahnya sebagai berikut :

1. Sistem hanya mendeteksi ada tidaknya ketersediaan tempat parkir di tempat parkir Fakultas Teknologi Industri Universitas Islam Sultan Agung.
2. Sistem hanya memberikan informasi kepada pengelola tempat parkir.
3. Sistem hanya mendeteksi ketersediaan tempat parkir untuk kendaraan mobil.
4. Pengambilan video untuk sistem deteksi ini hanya menggunakan kamera *webcam*.

1.4 Tujuan Tugas Akhir

Tujuan dari penelitian ini adalah untuk merancang sistem deteksi tempat parkir menggunakan metode Mask R-CNN serta dapat mendeteksi objek dengan akurat.

1.5 Manfaat

Manfaat yang diharapkan dari sistem ini adalah :

1. Mengurangi kemacetan apabila sedang mencari tempat parkir.
2. Meningkatkan efisiensi waktu saat mencari tempat parkir yang kosong.

1.6 Sistematika Penulisan

Sistematika penulisan yang akan digunakan oleh penulis dalam sebuahpembuatan laporan tugas akhir adalah sebagai berikut:

BAB 1: PENDAHULUAN

Pada bab ini penulis mengutarakan latar belakang pemilihan judul, Rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2: TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini memuat penelitian-penelitian sebelumnya dan dasar teori yang berguna untuk membantu sebuah penulis untuk memahami bagaimana konsep kerja dari sistem deteksi ketersediaan parkir menggunakan *Mask* R-CNN

BAB 3: METODE PENELITIAN

Bab ini mengungkapkan proses tahapan-tahapan penelitian dimulai dari perancangan sistem, alur kerja sistem, dan pengujian sistem.

BAB 4: HASIL PENELITIAN

Pada bab ini penulis mengungkapkan hasil penelitian yakni implementasi sistem pada komputer dengan bantuan perangkat keras menggunakan kamera *webcam* sebagai alat untuk mengambil data tempat parkir yang kemudian proses deteksinya akan menggunakan *Mask* R-CNN.

BAB 5: KESIMPULAN DAN SARAN

Bab ini penulis memaparkan kesimpulan daripada proses penelitian dari awal hingga akhir dan saran yang akan dikembangkan.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Adapun beberapa penelitian sebelumnya yang menjadi bahan pertimbangan dan sumber referensi yang berhubungan dengan judul penelitian ini adalah sebagai berikut:

Penelitian yang pertama bertujuan mendeteksi ketersediaan slot parkir dengan fokus mengimplementasikan metode *Histogram of Oriented Gradients* dan *Support Vector Machine* untuk mengetahui performa dari metode tersebut.

pada level pelatihan model SVM, ekstraksi ciri HOG digunakan untuk mengumpulkan fitur citra dan untuk pengujiannya digunakan untuk mengambil ciri citra sebagai informasi masukkan pengujian. *K-fold cross validation* merupakan metode yang ada pada klasifikasi SVM sebagai cara mendapatkan model yang terbaik. Peforma yang terbaik didapatkan pada pengujian data utama menggunakan RBF kernel dengan akurasi tertinggi pada kondisi *cloudy* sebesar 99.6% (Putra & Candradewi, 2017).

Selanjutnya adalah penelitian yang menggunakan sebuah sistem embedded untuk mendeteksi ketersediaan slot parkir yang terdiri dari modul kamera V2 Raspberry Pi dan Raspberry Pi 3. Pada sistem tersebut diterapkan metode morfologi dan *background subtraction* sehingga sistem dapat mendeteksi jumlah slot parkir dan ketersediaan tiap slot parkir sesuai dengan keadaan sebenarnya. Berdasarkan pengujian dari penelitian tersebut, didapatkan dua hasil dari tempat parkir dengan kondisi garis parkir yang berbeda yaitu lurus dan serong, sehingga didapatkan tingkat akurasi sebesar 100% (Maulana dkk. 2018).

Kemudian terdapat penelitian untuk mendeteksi lokasi parkir kosong dengan melakukan simulasi menggunakan miniatur parkir mobil. Penelitian ini memanfaatkan metode ekstraksi ciri objek dimana peneliti menghitung nilai *eccentricity* dari objek pada slot parkir dan objek mobil sehingga dapat memberikan kriteria apakah slot parkir sudah terisi atau belum. Penelitian tersebut menghasilkan aplikasi desktop yang digunakan untuk menguji deteksi

slot tempat parkir, serta situs web untuk menampilkan informasi ketersediaan tempat parkir (Rahman & Dafitri, 2019).

Berikutnya merupakan penelitian yang bertujuan untuk mendeteksi ketersediaan tempat parkir namun dengan proses deteksi yang harus lebih cepat. Oleh karena itu peneliti menggunakan Cuda Dan *Modified* Yolo. CUDA yang merupakan platform komputasi paralel dan model pemrograman dikombinasikan dengan *Modified* Yolo (M-Yolo) untuk mengolah citra sekaligus mengolah data untuk mendeteksi citra mobil dan jumlah mobil secara parallel sehingga proses deteksi lebih cepat. Hasil uji coba menunjukkan bahwa dengan menggunakan GPU dibandingkan dengan CPU dapat mempercepat waktu komputasi rata- rata sebesar 0,179 detik dengan rata-rata akurasi sebesar 100% (Jupiyandi dkk., 2019).

Yang terakhir adalah penelitian untuk mendeteksi hunian parkir dimana peneliti bertujuan untuk menemukan model *Convolutional Neural Network* (CNN) yang terbaik dalam deteksi ketersediaan tempat parkir. Penelitian tersebut memanfaatkan beberapa model CNN seperti AlexNet, LiteAlexNet, VGG16, dsb yang digunakan untuk mendeteksi ketersediaan tempat parkir. Hasil dari penelitian ini berupa aplikasi android yang menampilkan ketersediaan tempat parkir serta disimpulkan bahwa model arsitektur terbaik ada pada modifikasi VGG16 dengan akurasi pada *training set* sebesar 99,75% dan *loss* sebesar 0,0090 (Assidhiqi, 2021)

2.2 Dasar teori

2.2.1 Tempat Parkir

Tempat parkir merupakan suatu fasilitas yang banyak ditemui di tempat-tempat tertentu seperti pusat perbelanjaan, bisnis maupun instansi. Sistem parkir yang terdapat di Indonesia itu sendiri ada dua, yaitu sistem parkir tradisional dan modern. Di zaman sekarang ini sistem parkir modern dimana informasi dari pengguna parkir serta informasi tentang ketersediaan ruang parkir yang diolah dalam sebuah computer merupakan hal yang sudah semestinya diterapkan pada tempat-tempat parkir saat ini.

Namun masih banyak tempat parkir yang masih menggunakan sistem tradisional seperti memberikan karcis serta melayani pengguna parkir secara manual. Hal ini dapat menyebabkan masalah bagi mereka yang ingin memarkir kendaraannya karena harus berkeliling untuk mencari tempat parkir yang kosong. Sulitnya mencari tempat parkir juga dapat menyebabkan kemacetan karena rata-rata mobil bergerak lebih lambat.

Gambar 2.1 Tempat Parkir

2.2.2 Citra Digital

Representasi dari suatu objek ataupun benda disebut dengan citra. Citra terdiri dari kombinasi antara titik, garis, bidang dan warna. Sebuah citra dapat diartikan sebagai fungsi kontinu dari intensitas cahaya pada suatu bidang dua dimensi yang disimbolkan dengan f(x, y) berukuran M baris dan N kolom, dengan amplitudo f di titik koordinat (x, y) dan koordinat spasialnya adalah x dan y yang dinamakan intensitas atau tingkat keabuan dari sebuah citra pada titik tersebut secara sistematis (Limbong, 2020).

Digitalisasi citra atau proses representasi dari suatu citra (kontinu) secara numerik dengan nilai-nilai diskrit. Merupakan proses pengolahan citra menggunakan komputer yang dilakukan dengan konversi citra analog ke citra digital.

2.2.3 Open CV

Salah satu *software* pustaka yang ditujukan untuk pengolahan citra dinamis secara *real-time*, dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez adalah OpenCV (*Open Source Computer Vision Library*). *Software* ini dikeluarkan berdasarkan lisensi permisif BSD yang lebih bebas dari pada GPL,

serta memberikan kebebasan sepenuhnya untuk digunakan secara komersil tanpa perlu memperlihatkan kode sumbernya. OpenCV juga memiliki *user interface* yang mendukung sistem operasi Windows, Linux, Mac OS, iOS dan Android termasuk untuk bahasa pemrograman C++, C, Python dan Java. Difokuskan pada aplikasi *real-time* serta digunakan untuk efisiensi dalam komputasi merupakan desain utama dari OpenCV (Zein, 2018).

Gambar 2.2 OpenCV

2.2.4 Region Based Convolutional Neural Network

R-CNN merupakan bagian dari ConvNet/CNN (*Convolutional Neural Network*) yang merupakan sebuah algoritma *Deep Learning* yang dapat mengambil gambar input, menetapkan kepentingan (bobot dan bias yang dapat dipelajari) untuk berbagai aspek/objek dalam gambar serta dapat membedakan-bedakan objek yang satu dengan objek yang lain. Namun dalam sebuah gambar, terkadang terdapat beberapa objek yang harus ditemukan. Oleh karena itu pada kasus deteksi objek, diperlukan untuk menggambar beberapa kotak pembatas (*bounding box*) untuk mewakili berbagai objek yang ada dalam gambar. Tetapi akan sulit untuk menyelesaikan permasalahan ini dengan CNN karena panjang lapisan keluaran merupakan variabel (tidak konstan), ini karena jumlah kemunculan objek yang diinginkan tidak tetap (Gandhi, 2022).

Untuk mengatasi masalah pemilihan sejumlah besar objek/wilayah, Ross Girshick mengusulkan metode di mana ia menggunakan pencarian selektif untuk mengekstrak hanya 2000 wilayah dari suatu gambar dan ia menyebutnya

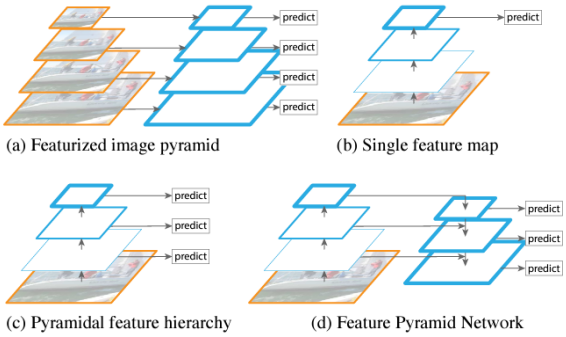
proposal wilayah (*region proposals*). Proposal 2000 wilayah ini dihasilkan menggunakan algoritma pencarian selektif (*selective search*) berikut :

1. Membuat sub-segmentasi awal, kemudian menghasilkan banyak kandidat daerah.
2. Menggunakan *Greedy Algorithm* untuk secara rekursif menggabungkan daerah yang sama menjadi yang lebih besar.
3. Menggunakan *region* yang dihasilkan untuk membuat proposal final kandidat wilayah.

Gambar 2.3 Cara kerja R-CNN

2.2.5 FPN

Feature Pyramid Network (FPN) merupakan arsitektur topdown dengan koneksi lateral yang dikembangkan untuk membangun peta fitur semantik level tinggi di semua skala. Arsitektur ini menunjukan perkembangan yang signifikan seperti ekstraktor fitur generic di beberapa penerapan.



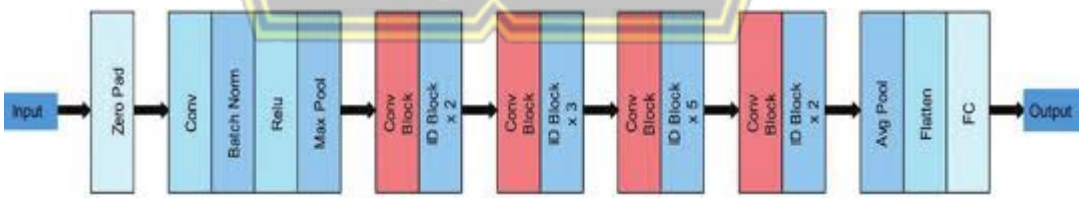
Gambar 2.4 Perkembangan FPN

Gambar (a) menggunakan piramida gambar untuk membangun fitur

piramid. Fitur-fitur tersebut dikomputasi pada setiap skala gambar secara individu sehingga prosesnya lambat. Pada gambar (b) system deteksi terbaru telah memilih untuk menggunakan fitur skala tunggal untuk deteksi yang lebih cepat. Gambar (c) merupakan alternatif untuk menggunakan kembali komputasi hirarki fitur pyramid oleh ConvNet, seolah-olah berupa unggulan pyramid gambar. FPN juga memiliki proses yang cepat seperti gambar (b) dan (c), tetapi lebih akurat. Pada gambar tersebut, peta fitur ditunjukkan dengan garis biru dan garis yang lebih tebal menunjukkan secara semantic fitur yang lebih kuat (Lin, Tsung-Yi dkk, 2017).

2.2.6 Resnet-101

Dengan menggunakan koneksi pintasan untuk menghindari masalah disperse gradien sekaligus memperdalam kedalaman jaringan, Resnet 101 adalah struktur jaringan yang diusulkan oleh Kaiming He dkk. Baik dari ResNet101 dan ResNet50 sangat sering digunakan dalam berbagai penelitian sebagai ekstraktor fitur dasar karena ia sangat sederhana dan praktis. ResNet memiliki peran penting dalam masalah visi komputer serta mewakili sisa jaringan. Ditunjukkan pada gambar 2.5, ResNet101 berisi 104 lapisan convolutional yang terdiri dari 33 blok lapisan, dan 29 kotak ini langsung digunakan di blok sebelumnya. Jaringan ini dilatih dengan 1000 kelas objek yang berdasar pada kumpulan data ImageNet.



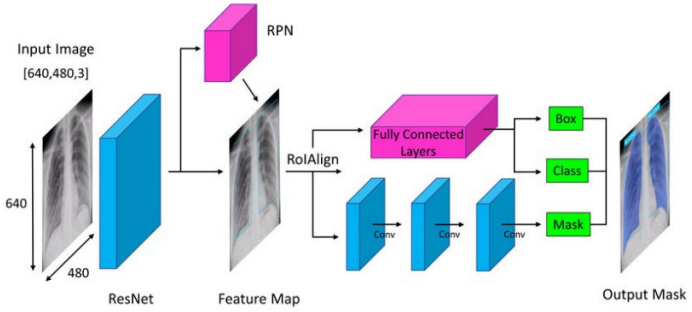
Gambar 2.5 Susunan arsitektur ResNet101

"Koneksi pintasan" merupakan metode koneksi yang digunakn oleh Resnet, dimana pada struktur resnet akurasi deteksi target dapat meningkat pesat dengan mempercepat pelatihan jaringan syaraf dalam yang sangat cepat. (Hanun dkk., 2023).

2.2.7 Mask-RCNN

Mask R-CNN, atau *Mask* RCNN, adalah *Convolutional Neural Network* (CNN) dan merupakan bagian dalam hal segmentasi gambar dan *instance segmentation*. *Mask* R-CNN dikembangkan di atas *Faster* R-CNN yang merupakan arsitektur dasar dari terbentuknya *Mask* R-CNN. Algoritma ini memiliki *output* cabang ketiga dari deteksi objek pada umumnya yaitu mengeluarkan topeng objek (*object mask*) sementara *Faster* R-CNN hanya memiliki dua keluaran untuk setiap usulan objek yaitu label kelas, dan *bounding-box*. Keluaran topeng objek membutuhkan ekstraksi tata letak spasial yang jauh lebih baik dari suatu objek, berbeda dari keluaran kelas dan *box* (Odemakinde, 2021).

Cara kerja awal dari Mask R-CNN dimulai dari pemrosesan gambar menggunakan *backbone* seperti ResNet dan FPN yang digunakan untuk mengekstrak fitur map. Selanjutnya fitur map akan diproses melalui *Region Proposed Network* (RPN) yang akan menghasilkan kandidat *Region of Interest* (RoI). Selanjutnya pada RoI align akan ditemukan area-area yang relevan sampai level piksel dimana hasil keluarannya akan dibagi menjadi dua. Hasil dari RoI *align* yang pertama akan diproses pada *Fully Connected Layer* yang dipakai untuk deteksi kepala. Proses tersebut digunakan untuk mengklasifikasikan pemberian label kelas dan juga kotak pembatas (*box*). Kemudian hasil yang kedua akan melalui proses percabangan *mask*, dimana hasil RoI *align* akan diproses terus menerus untuk mengahasilkan *binary mask* dimana hanya *mask* yang relevan dengan prediksi kelas saja yang akan dipakai. Untuk lebih jelasnya dapat dilihat pada gambar 2.6.



Gambar 2.6 Arsitektur Mask R-CNN

2.2.8 Confusion Matrix

Memperlihatkan jumlah data yang diklasifikasikan dengan benar dan yang salah oleh model dalam *machine learning* yang digambarkan pada untuk mengevaluasi peforma model klasifikasi biasa disebut dengan *Confusion matrix* (Hozairi dkk, 2021).

Tabel 2.1 *Confusion Matrix*

	Actually Positive	***Actually Negative***
Predicted Positive	*True Positive* (TP)	*False Positive* (FP)
Predicted Negative	*False Negative* (FN)	*True Negative* (TN)

Sesuai pada tabel 2.1 *confusion matrix* memiliki empat nilai utama yang menggambarkan hasil dari proses klasifikasi, yaitu:

1. *True Positive* (TP)**:** Nilai TP merupakan lokasi tempat parkir yang terisi dan juga diprediksi dengan benar oleh model sebagai tempat parkir yang terisi. Dalam konteksnya, ini jumlah tempat parkir terisi yang berhasil diidentifikasi dengan benar oleh model.
2. *True Negative* (TN)**:** Nilai TN adalah jumlah tempat parkir yang kosong dan diprediksi dengan benar oleh model sebagai tempat parkir kosong. Dalam konteksnya, ini adalah jumlah tempat parkir kosong yang berhasil diidentifikasi dengan benar oleh model.
3. *False Positive* (FP): Nilai FP adalah jumlah slot tempat parkir yang diprediksi oleh model sebagai terisi, namun sebenarnya slot tempat parkir sedang kosong. Dalam konteknya, ini adalah jumlah slot tempat parkir kosong yang keliru diidentifikasi sebagai slot tempat parkir terisi oleh model.
4. False *Negative* (FN): Nilai FN adalah jumlah tempat parkir yang diprediksi oleh model sebagai slot tempat parkir kosong tetapi sebanarnya slot tempat parkir terisi. Dalam konteknya, ini adalah jumlah tempat parkir terisi namun salah diidentifikasi oleh model sebagai tempat parkir kosong.

Dengan nilai-nilai ini, kita dapat melakukan berbagai perhitungan untuk mengevaluasi performa model, seperti:

1. *Accuracy* adalah proporsi dari prediksi yang benar, dihitung dengan rumus sebagai berikut :

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

2. *Precision* dalah proporsi dari prediksi positif yang benar Dalam konteks ini, presisi akan memberi tahu kita seberapa baik model dalam mengidentifikasi ketersediaan tempat parkir. Precision dihitung dengan rumus sebagai berikut

$$Precision = \frac{TP}{TP+FP}$$

3. *Recall* adalah proporsi dari data aktual yang benar-benar diidentifikasi sebagai positif oleh model, Dalam konteks ini, *recall* akan memberi tahu kita sejauh mana model mampu mendeteksi *chest x-ray* tuberkulosis. *Recall* dihitung dengan rumus sebagai berikut

$$Recall = \frac{TP}{TP+FN}$$

4. *F1-Score* adalah metrik gabungan yang memadukan *precissioni* dan *recall* dalam memberikan perspektif yang lebih luas dan komprehensif tentang performa model. *F1-Score* dihitung dengan rumus sebagai berikut

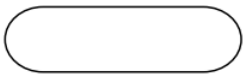
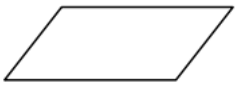
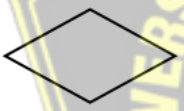
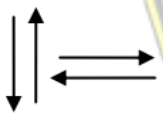
$$F1\text{-}Score = \frac{2 * Precission * Recall}{Precission + Recall}$$

2.2.9 Flowchart

Proses pemecahan masalah yang ditulis dengan simbol-simbol tertentu disebut dengan *Flowchart*. *Flowchart* ini akan menampilkan alur dalam program secara logika. Flowchart diperlukan sebagai panduan dan sebelum pemahaman yang lebih baik dari komponen-komponen *flowchart*, tidak hanya digunakan

sebagai alat komunikasi saja (Khesya, 2021). Pada tabel 2.2. merupakan bentuk beserta kegunaan dari simbol-simbol yang sering digunakan untuk menggambarkan suatu algoritma dalam bentuk diagram alir.

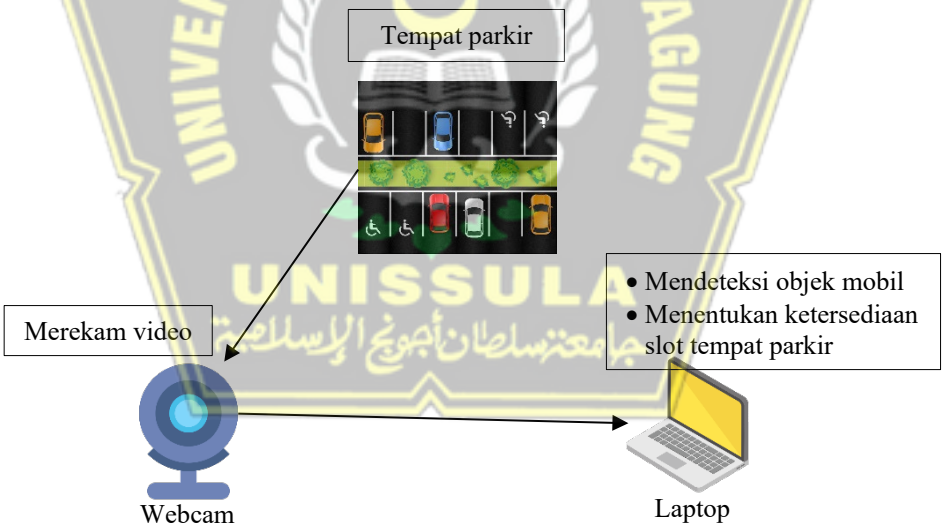
Tabel 2.2 Simbol *flowchart*

No	Simbol	Nama	Fungsi Simbol
1		*"Terminal"*	Awal atau akhir suatu program (Prosedur).
2		*"Output/Input"*	Proses input atau output terlepas dari jenis perangkat.
3		*"Procces"*	Proses operasional computer.
4		"*Decision*"	Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan, ya/tidak.
5		*"Predefined Process"*	Mewakili ketentuan penyimpanan untuk diproses untuk memberikan awal harga.
6		*"Flow"*	Menyatakan jalannya arus suatu proses.

BAB III
METODE PENELITIAN

3.1 Deskripsi Sistem

Hasil dari penelitian ini yaitu sebuah program untuk mendeteksi ketersediaan tempat parkir yang menampilkan informasi ketersediaan tempat parkir pada video *real-time* di tempat parkir FTI UNISSULA. Sistem ini dijalankan pada sebuah komputer pribadi yang menggunakan kamera *webcam* untuk melakukan proses observasi, yang kemudian diproses untuk mendeteksi objek mobil menggunakan algoritma Mask R-CNN serta menghitung nilai IoU untuk mengklasifikasikan apakah slot parkir tersedia atau tidak. Semua data tersebut akan ditampilkan pada monitor pada laptop maupun komputer yang juga akan menampilkan informasi mengenai jumlah tempat parkir yang masih tersedia.



Gambar 3.1 Rancangan sistem

Pada gambar 3.1 kamera digunakan untuk mengambil video tempat parkir FTI UNISSULA yang ditempatkan pada gedung FTI lantai tiga. Hasil video yang didapat akan diproses pada laptop untuk dideteksi objek mobil yang ada serta menentukan slot parkir yang terisi maupun kosong pada tempat parkir FTI UNISSULA.

3.2 Kebutuhan Sistem

Perangkat kerja dan perangkat lunak yang dibutuhkan pada implementasi sistem deteksi ketersediaan tempat parkir ini adalah sebagai berikut.

3.2.1 Laptop

Perangkat keras utama yang digunakan dalam penelitian adalah laptop dengan merk ASUS Vivobook seri X441UB yang merupakan laptop milik penulis. Laptop ini digunakan mulai dari melatih dan mengevaluasi model Mask-RCNN hingga menjalankan program deteksi ketersediaan tempat parkir. Untuk lebih jelasnya dapat dilihat pada tabel 3.1 untuk spesifikasi lengkap.

Tabel 3.1 Spesifikasi laptop

CPU	Intel® Core™ i3-6006U 2.0GHz
Integrated GPU	Intel HD Graphic 520 128 MB
Dedicated GPU	NVIDIA GeForce MX110 2 GB
RAM	8 GB 2133 MHz DDR 4
Penyimpanan	SSD V-Gen 512 GB
Resolusi layar	1366x768p
Konektivitas	Integrated 802.11b/g/n, Bluetooth® 4.0
Interface	1 x COMBO audio jack, 1 x VGA port, 1 x USB 3.1 Type C port(s), 1 x Type-C USB 3.0 (USB 3.1 Gen 1), 1 x USB 2.0 port(s), 1 x RJ45 LAN jack for LAN insert, 1 x HDMI, 1 x AC adapter plug

3.2.2 *Webcam*

Webcam Logitech C270 digunakan untuk memperoleh data yang video tempat parkir FTI UNISSULA yang nantinya akan diproses oleh Laptop ASUS X441UB untuk diaplikasikan pada sistem. Pada Webcam Logitech C270 memiliki spesifikasi yang ditunjukan pada tabel 3.2 halaman 17.

Tabel 3.2 Spesifikasi *Webcam*

Resolusi	1080px720p
Kecepatan	30 *frame rate per second*
Permukaan Penglihatan	55°
Fokus Tetap	Lensa Plastik
Mikrofon Mono Omnidirectional	Terintegrasi
Jenis Teknologi	Teknologi Peredam Kebisingan
Koreksi Cahaya	Otomatis
Klip Pemasangan	*Universal*
Pasang dan Mainkan melalui	USB 2.0 Tipe-A
Kompatibel	Windows, macOS, dan Chrome

3.2.3 Python 3.7.9

Python adalah bahasa pemrograman tingkat tinggi yang memiliki sintaksis yang sederhana untuk dibaca. Selain mendukung berbagai aplikasi, ia serbaguna dan dapat digunakan di berbagai platform. Penelitian ini menggunakan Python versi 3.7.9 karena bersifat *open source*, komunitasnya besar, dan ada banyak sumber daya online. Python menjadi pilihan populer untuk pengembangan perangkat lunak karena fokusnya pada produktivitas, manajemen memori otomatis, dan integrasi mudah dengan bahasa lain.

3.2.4 Library untuk Mask-RCNN

Pada implementasi Mask-RCNN oleh Matterport (Abdulla, 2017), terdapat beberapa *library* python yang dibutuhkan agar algoritma ini dapat digunakan. Daftar *library* yang dibutuhkan berada di file requirements.txt yang ada pada repositori GitHub Matterport Mask-RCNN. Berikut ini daftar library yang dibutuhkan.

- Numpy

  Paket dasar untuk komputasi ilmiah yang digunakan pada Python disebut dengan numpy. Ia menyediakan dukungan untuk array dan matriks multidimensi yang besar, bersama dengan kumpulan fungsi matematika untuk beroperasi pada *array* ini. Numpy sangat diperlukan pada algoritma Mask R-CNN untuk menangani komputasi numerik.

- Scypy

  Scipy adalah *library* Python yang digunakan untuk komputasi ilmiah dan teknis. Library ini dibangun di atas Numpy dengan menambahkan kumpulan algoritma dan perintah tingkat tinggi untuk manipulasi dan analisis data. Scipy sering digunakan untuk optimasi, integrasi, dan operasi matematika tingkat lanjut lainnya.

- Pillow

  Pillow adalah fork Python Imaging Library (PIL) yang mendukung pembukaan, manipulasi, dan penyimpanan berbagai format file gambar. Pillow digunakan untuk tugas pemrosesan awal gambar seperti mengubah ukuran, memotong, dan mengubah gambar sebelum dimasukkan ke jaringan saraf.

- Cython

  Cython adalah bahasa pemrograman yang membuat penulisan ekstensi C untuk Python semudah Python itu sendiri. Cython digunakan untuk mengkompilasi kode Python ke dalam C untuk meningkatkan kinerjanya. Ini bisa sangat berguna dalam mengoptimalkan bagian Mask R-CNN yang intensif lewat komputasi.

- Matplotlib

  Matplotlib adalah *library* plot untuk bahasa pemrograman Python dan ekstensi matematika numerik dari Numpy. Matplotlib digunakan untuk memvisualisasikan hasil model, seperti memplot objek yang terdeteksi dan *mask* yang ada pada gambar.

- Scikit-Image

  Scikit-Image adalah *library* pemrosesan gambar sumber terbuka untuk Python. Ia menyediakan kumpulan algoritma untuk pemrosesan gambar, termasuk

fungsi untuk membaca, mengubah, dan menganalisis gambar. Ini berguna untuk gambar yang sebelum dan sesudah melakukan pemrosesan di Mask R-CNN.

- TensorFlow

  TensorFlow adalah *framework* pembelajaran mendalam sumber terbuka yang dikembangkan oleh Google. TensorFlow adalah *framework* utama yang digunakan untuk mengimplementasikan dan melatih model Mask R-CNN. Ini menangani komputasi kompleks yang diperlukan untuk pembelajaran mendalam.

- Keras

  Keras adalah API jaringan saraf tingkat tinggi, ditulis dengan Python dan mampu berjalan di atas TensorFlow, Microsoft Cognitive Toolkit, Theano, atau PlaidML. Keras menyederhanakan pembuatan dan pelatihan model pembelajaran mendalam. Dalam konteks Mask R-CNN, library ini digunakan untuk mendefinisikan dan melatih lapisan dan arsitektur jaringan saraf.

- OpenCV-Python

  OpenCV adalah library perangkat lunak visi komputer dan pembelajaran mesin sumber terbuka. OpenCV digunakan untuk tugas pemrosesan gambar dan video, seperti membaca dan menulis gambar, transformasi gambar, dan menggambar kotak pembatas dan masker pada gambar.

- H5py

  H5py adalah antarmuka Python ke format data biner HDF5. H5py digunakan untuk menyimpan dan memuat data numerik dalam jumlah besar secara efisien. Di Mask R-CNN, *library* ini sering digunakan untuk menyimpan bobot model yang dilatih.

- Imgaug

  Imgaug adalah *library* untuk augmentasi gambar dalam eksperimen pembelajaran mesin. Imgaug digunakan untuk menambah gambar pelatihan guna meningkatkan ketahanan dan generalisasi model Mask R-CNN. Teknik augmentasi ini meliputi membalik, memutar, menskalakan, dan transformasi lainnya.

- IPython

  IPython menyediakan lingkungan komputasi interaktif untuk Python. IPython sering digunakan dalam lingkungan penelitian dan pengembangan untuk menjalankan dan men-*debug* kode secara interaktif. Library ini meningkatkan produktivitas dengan menyediakan alat yang kaya untuk pemrograman interaktif.

3.2.5 CUDA dan cuDNN

model pemrograman dan platform komputasi paralel yang dibangun oleh NVIDIA untuk komputasi umum pada *Graphics Processing Unit* (GPU). Ini membuat pengembang dapat memanfaatkan kekuatan pemrosesan paralel dari GPU NVIDIA untuk mempercepat tugas-tugas komputasi. Pada penelitian ini penulis menggunakan CUDA versi 10.1 yang merupakan versi yang sesuai dengan spesifikasi GPU laptop yang ada, yaitu NVIDIA MX110. Aplikasi ini dapat diunduh secara gratis pada halaman resmi NVIDIA.

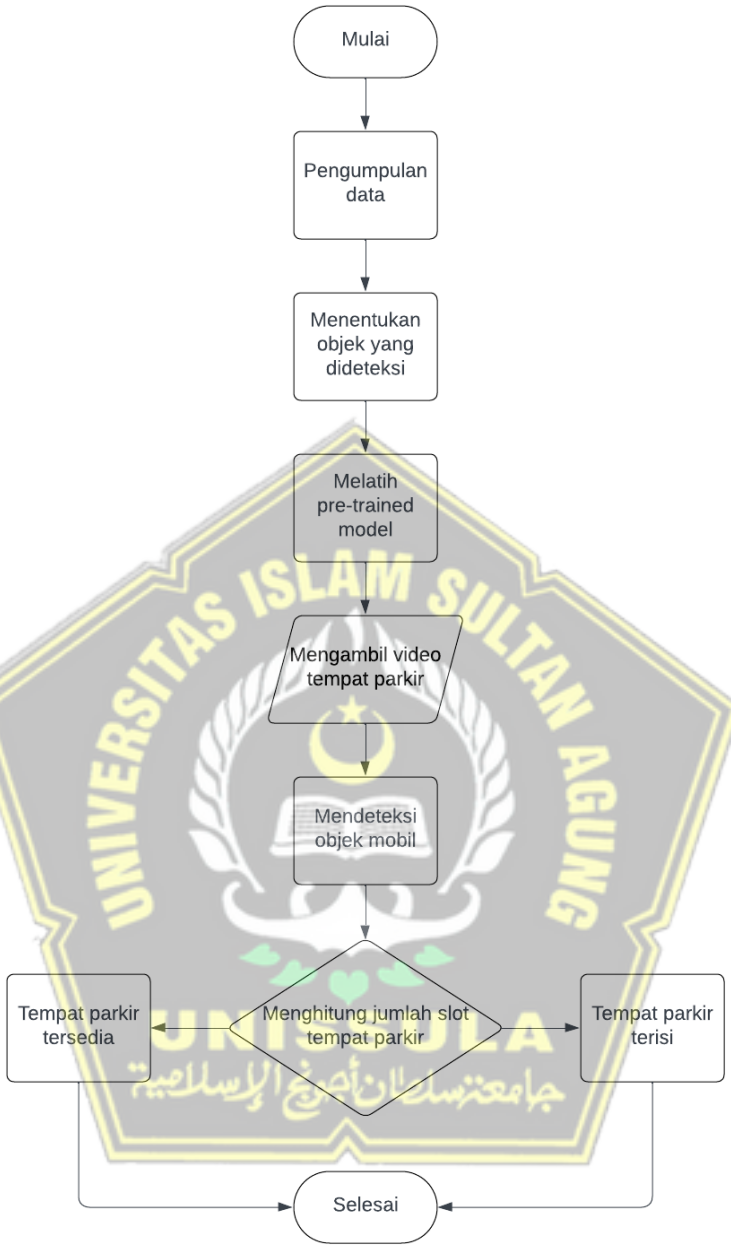
Gambar 3.2 Situs web NVIDIA

Penggunaan CUDA dalam Mask R-CNN adalah untuk menyediakan infrastruktur yang diperlukan untuk menjalankan komputasi paralel pada GPU NVIDIA. Saat melatih atau menjalankan inferensi dengan model pembelajaran mendalam, CUDA dapat menangani distribusi komputasi di seluruh inti GPU.

cuDNN (CUDA *Deep Neural Network Library*) adalah *library* GPU yang dipercepat untuk *deep neural network* yang dikembangkan oleh NVIDIA. cuDNN dibangun di atas CUDA dan menyediakan implementasi yang sangat dioptimalkan dari rutinitas standar yang digunakan dalam pembelajaran mendalam. Pada penerapannya, cuDNN meningkatkan kinerja operasi pembelajaran mendalam secara khusus. Untuk Mask R-CNN, yang melibatkan penggunaan ekstensif dari lapisan konvolusional, operasi *pooling*, dan primitif jaringan saraf lainnya, cuDNN menyediakan implementasi yang dioptimalkan yang membuat model berjalan lebih cepat.

3.3 Analisis Sistem

Pada sistem ini akan diterapkan metode Mask-RCNN dengan backbone Resnet101 dan FPN untuk mendeteksi mobil dan menentukan ketersediaan tempat parkir. Sistem ini akan dibangun menggunakan dua perangkat yaitu *webcam* yang akan digunakan untuk pengambilan data dan pengujian sistem yang akan dibuat, serta laptop yang digunakan untuk merancang sistem deteksi ketersediaan tempat parkir dan juga menjalankan program tersebut yang nantinya akan menampilkan informasi mengenai ketersediaan tempat parkir di FTI UNISSULA. Berikut *Flowchart* alur sistem dari sistem deteksi ketersediaan tempat parkir ini beserta penjelasannya pada gambar 3.3 halaman 22.

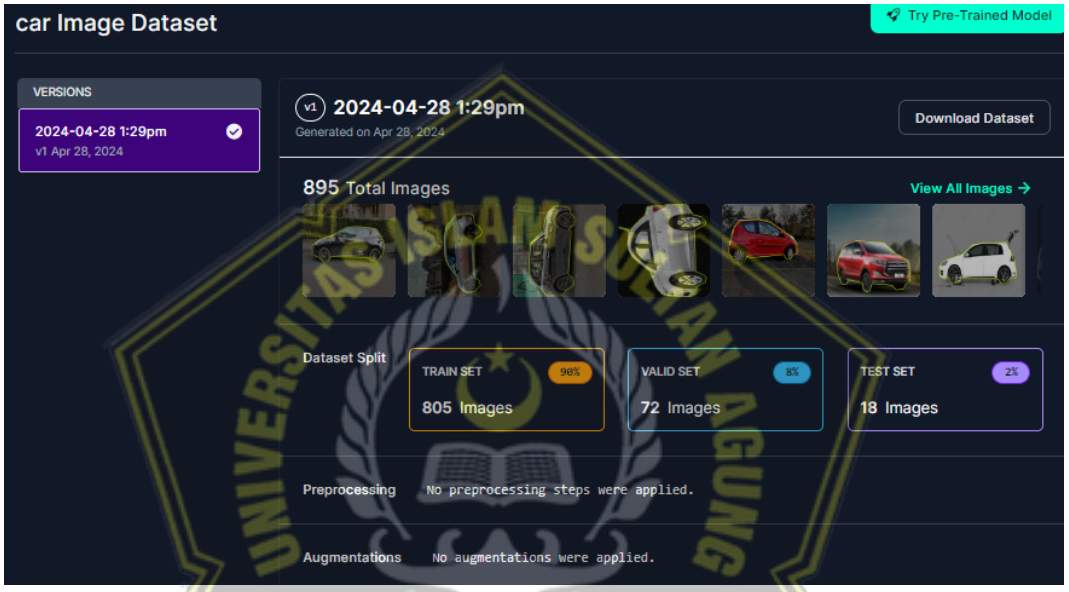


Gambar 3.3 *Flowchart* alur sistem

3.3.1 Pengumpulan Data

Pertama-tama dilakukan pengumpulan data terlebih dahulu, mulai dari data untuk melatih model hingga data untuk melakukan pengujian. Dataset yang digunakan untuk melatih model Mask R-CNN didapatkan melalui internet pada situs web Roboflow. Roboflow sendiri merupakan platform yang dirancang khusus untuk membantu AI *engineer* untuk memperoleh dan memproses dataset

untuk pengaplikasian proyek *computer vision*. Pada halaman resmi Roboflow, penulis mengambil car image dataset yang merupakan dataset yang berisi kumpulan gambar mobil (Aa, 2024). Dataset ini menggunakan anotasi berstandar COCO dan juga hanya memiliki satu label saja yaitu car. Dataset ini telah dibagi menjadi tiga yaitu data training 805 gambar(90%), validasi 72 gambar(8%), dan testing 18 gambar(2%). Pada gambar 3.4 merupakan situs web Roboflow berisi *car Image Dataset* yang digunakan pada penelitian ini.



Gambar 3.4 Situs Roboflow untuk dataset *car*

3.3.2 Menetukan Objek Deteksi

Objek yang digunakan untuk deteksi pada system ini hanya satu objek saja yaitu mobil. Dataset yang digunakan pada penelitian ini (*car Image Dataset*)pun hanya memiliki satu kelas, yaitu car.

3.3.3 Melatih *Pre-Trained* Model

Selanjutnya adalah mempersiapkan sebuah *pre-trained* model *Mask* R-CNN yang akan dilatih menggunakan *car Image Dataset*. Pre-trained weights yang digunakan merupakan model Mask R-CNN yang telah dilatih menggunakan *dataset* COCO tahun 2017 yang terbagi menjadi 118.000 data latih, 5.000 data validasi, dan 41.000 data tes dimana *dataset* ini menyediakan

berbagai macam data latih untuk melatih model agar dapat mendeteksi berbagai objek, salah satunya data kendaraan rode empat untuk mendeteksi objek seperti mobil, truk, dan bus. Disini *pre-trained* model akan dilatih kembali dengan *car Image Dataset* agar akurasi dari deteksi objek mobil semakin akurat.

Gambar 3.5 Proses *training* model

Gambar 3.5 merupakan proses dalam melakukan training model. *Dataset* yang digunakan adalah *car Image Dataset* yang kemudian dilatih dengan metode Mask-RCNN pada pre-trained model. Model yang berhasil dilatih diberi nama `trainedcar` dengan ekstensi .h5. Pelatihan untuk model *Mask* R-CNN ini berguna untuk mendeteksi objek-objek mobil yang berada di tempat parkir FTI UNISSULA.

3.3.4 Mengambil Video tempat parkir

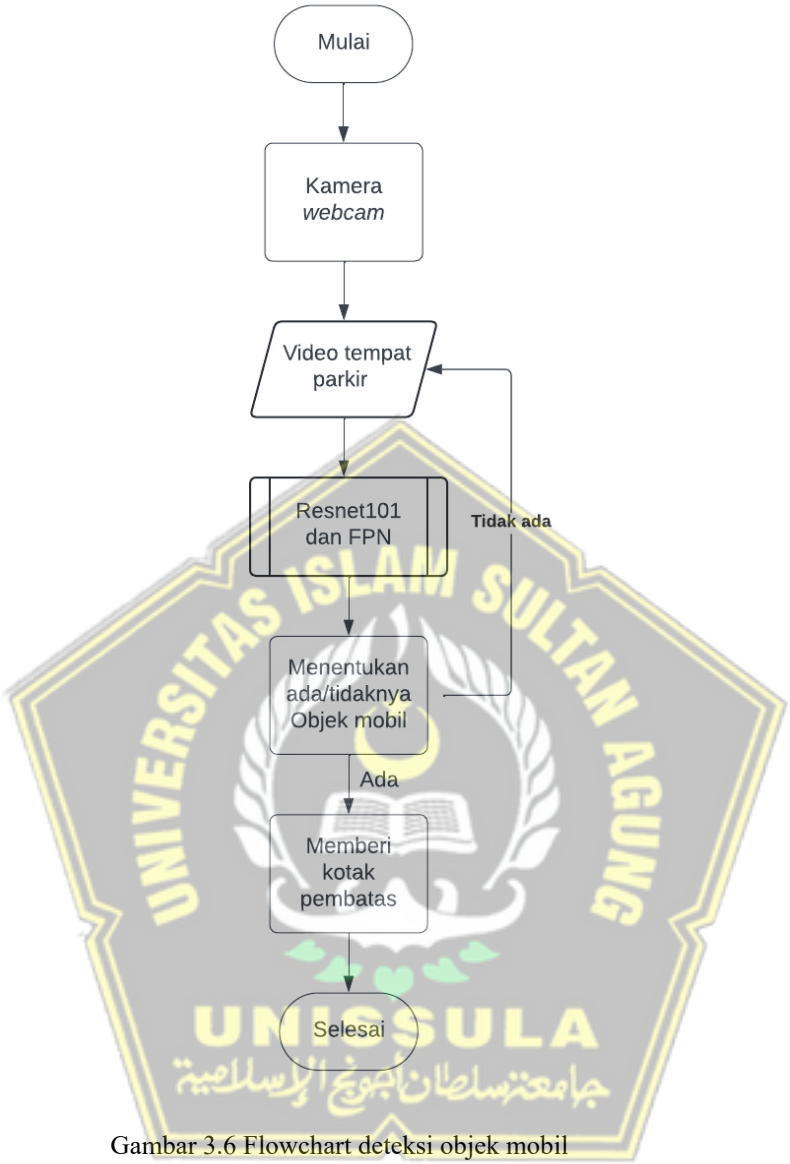
Selain data untuk melatih model, diperlukan juga data untuk melakukan pengujian. Data yang dibutuhkan untuk melakukan pengujian adalah data

rekaman video dari tempat parkir Fakultas Teknologi Industri Universitas Islam Sultan Agung yang diambil dari Gedung FTI lantai tiga. Video ini diambil menggunakan perangkat kamera *webcam* Logitech C270, dimana data ini akan dipakai untuk menentukan jumlah tempat parkir yang tersedia di FTI UNISSULA.

3.3.5 Menghitung Jumlah Slot Tempat Parkir

Pada saat program dijalankan untuk melakukan uji, program akan langsung mendeteksi terlebih dahulu objek mobil yang ada pada video tempat parkir. Objek mobil yang berhasil terdeteksi masing-masing akan diberikan kotak pembatas, dan lokasi awal kotak pembatas yang terdeteksi akan disimpan. Dengan begitu kotak pembatas yang ditampilkan pada program deteksi ketersediaan tempat parkir akan terbagi menjadi dua, yaitu kotak pembatas awal saat mobil berhasil dideteksi (kotak pembatas disimpan dan tidak bergerak) dan kotak pembatas yang melekat pada objek mobil yang bergerak.

Gambar 3.6 halaman 26 merupakan *flowchart* proses deteksi objek mobil yang ada pada tempat parkir FTI UNISSULA. Berdasarkan gambar tersebut, *webcam* akan mengambil video tempat parkir Fakultas Teknologi Industri UNISSULA. Selanjutnya video tempat parkir tersebut akan dikomputasikan menggunakan Mask R-CNN dimana metode ini menggunakan *backbone* Resnet101 dan FPN. Kedua *backbone* ini digunakan untuk ekstraksi fitur dan representasi fitur multi-skala yang mana Resnet101 mengekstrak representasi fitur yang kaya dari gambar/video masukan dan FPN menghasilkan peta fitur pada berbagai skala serta meningkatkan kinerja lokalisasi dan segmentasi model Mask R-CNN. Dari proses tersebut akan dideteksi ada tidaknya objek mobil. Jika tidak terdapat objek mobil maka akan dilihat kembali pada video tersebut untuk dideteksi apakah ada/tidak objek mobil dalam setiap *frame* video. Apabila objek mobil ditemukan maka akan diberikan kotak pembatas (*bounding boxes*) pada objek mobil tersebut.



Gambar 3.6 Flowchart deteksi objek mobil

Intersection Over Unit merupakan istilah yang digunakan untuk menggambarkan/memberikan nilai tingkat tumpang tindih dari dua buah kotak pembatas. IoU membagi area dari dua buah kotak yang saling berpotongan dengan area dari gabungan dua buah kotak. Nilai dari IoU sendiri berada diantara nol sampai satu.

Untuk menghitung IoU, pada penelitian ini digunakan dua buah kotak yaitu kotak pembatas objek mobil yang terdeteksi di awal mulai program dan kotak pembatas objek mobil yang melekat pada mobil itu sendiri. Ketika kotak pembatas antara dua buah objek tersebut saling tumpang tindih maka nilai IoU

akan berubah dan ketika nilai IoU mendekati nilai satu (nilai IoU tinggi) maka slot pada tempat parkir akan dikategorikan sebagai “sudah terisi”. Begitu juga sebaliknya, jika dua buah kotak tersebut tidak tumpang tindih ataupun saling tumpang tindih tetapi IoU bernilai rendah, maka slot parkir tersebut dikategorikan sebagai “belum terisi/kosong”.

```
if max_IoU_overlap < 0.15:
    # Parking space not occupied! Draw a green box around it
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 3)
    # Flag that we have seen at least one open space
    free_space = True
    if free_space ==True:
        space_count += 1
    else :
        space_count -= 1
else:
    # Parking space is still occupied - draw a red box around it
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 1)
```

Gambar 3.7 Kode untuk menghitung IoU

BAB IV
HASIL DAN ANALISIS PENELITIAN

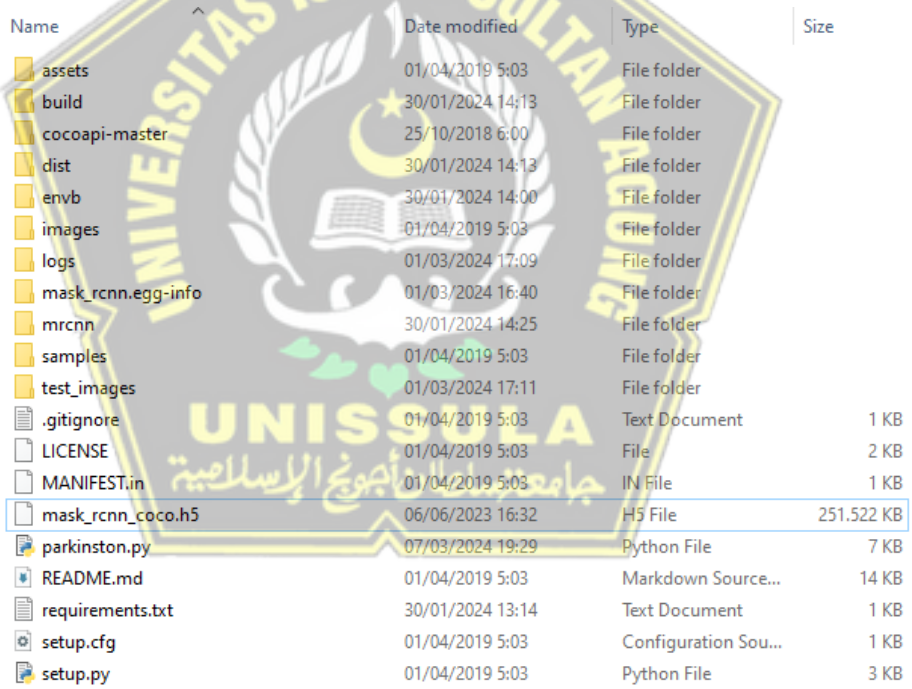
4.1 Implementasi Sistem

4.1.1 Instalasi Kebutuhan Mask-RCNN

Untuk menggunakan metode Mask-RCNN dibutuhkan beberapa langkah-langkah agar metode tersebut dapat digunakan.

a) *Cloning* Github *Repository* Mask-RCNN

Pertama-tama unduh terlebih dahulu github *repository* Mask-RCNN oleh Matterport pada tautan *github.com/matterport/Mask_RCNN.* Setelah mengunduh, ekstrak file Mask_RCNN-master pada penyimpanan di komputer sehingga menampilkan kumpulan file dan folder seperti pada gambar 4.1.



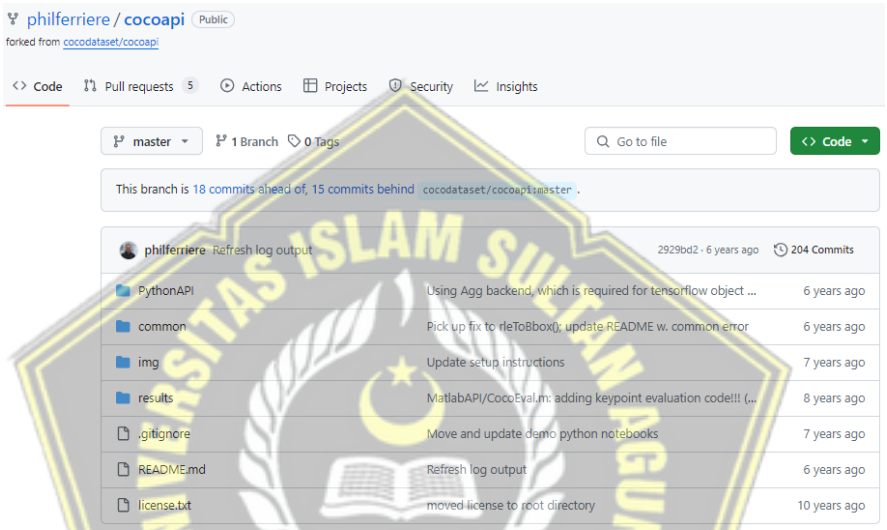
Gambar 4.1 hasil ekstraksi *file* maskrcnn-master.zip

Setelah berhasil diekstrak, anda dapat mengubah nama folder Mask_RCNN-master. Folder tersebut saya ubah namanya menjadi sparkir. Selanjutnya unduh *pre-trained* weights pada *repository* tersebut yang menggunakan *dataset* dari MS COCO. Untuk penelitian ini penulis menggunakan Mask-RCNN versi 2.0 yang

tersedia pada tautan *github.com/matterport/Mask_RCNN/releases*.

Gambar 4.2 *Pre-trained weights* mask r-cnn

Kemudian masih pada *repository* tersebut, unduh file pycocotools pada tautan *github.com/philferriere/cocoapi*.



Gambar 4.3 Repositori Github cocoapi

Ekstrak *file* cocoapi-master.zip pada folder sparkir yang terdapat pada komputer.

b) Instalasi requirement.txt dan Mask R-CNN

File requirement.txt berisi beberapa *library* yang dibutuhkan untuk menjalankan Mask R-CNN pada *virtual environment* envb yang sebelumnya telah dibuat terlebih dahulu.

```
Collecting numpy==1.20.3
  Using cached numpy-1.20.3-cp37-cp37m-win_amd64.whl (13.6 MB)
Collecting scipy==1.4.1
  Using cached scipy-1.4.1-cp37-cp37m-win_amd64.whl (30.9 MB)
Collecting Pillow==8.4.0
  Using cached Pillow-8.4.0-cp37-cp37m-win_amd64.whl (3.2 MB)
Collecting cython==0.29.24
  Using cached Cython-0.29.24-cp37-cp37m-win_amd64.whl (1.6 MB)
Collecting matplotlib
  Using cached matplotlib-3.5.2-cp37-cp37m-win_amd64.whl (7.2 MB)
Collecting scikit-image==0.16.2
  Using cached scikit_image-0.16.2-cp37-cp37m-win_amd64.whl (25.7 MB)
Collecting tensorflow==2.2.0
```

Gambar 4.4 Proses instalasi *library*

Pada gambar 4.4 merupakan proses pengunduhan dan instalasi *library* yang ada dalam *file* requirement.txt (daftar *library* ada pada bab iii kebutuhan sistem).

Setelah selesai melakukan instalasi *file* requirement.txt, selanjutnya adalah melakukan instalasi Mask R-CNN itu sendiri. Instalasi ini menggunakan *file* setup.py, yang merupakan *file* berisi kode perintah untuk melakukan instalasi *library* Mask-RCNN dengan memuat terlebih dahulu *library* yang sudah diinstal sebelumnya. Gambar 4.5 merupakan output proses saat melakukan instalasi Mask R-CNN.

```
running egg_info
creating mask_rcnn_tf2.egg-info
writing mask_rcnn_tf2.egg-info\PKG-INFO
writing dependency_links to mask_rcnn_tf2.egg-info\dependency_links.txt
writing top-level names to mask_rcnn_tf2.egg-info\top_level.txt
writing manifest file 'mask_rcnn_tf2.egg-info\SOURCES.txt'
reading manifest file 'mask_rcnn_tf2.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
adding license file 'LICENSE'
writing manifest file 'mask_rcnn_tf2.egg-info\SOURCES.txt'
installing library code to build\bdist.win-amd64\egg
running install_lib
running build_py
creating build
creating build\lib
creating build\lib\mrcnn
copying mrcnn\config.py -> build\lib\mrcnn
copying mrcnn\model.py -> build\lib\mrcnn
copying mrcnn\model_temp.py -> build\lib\mrcnn
copying mrcnn\parallel_model.py -> build\lib\mrcnn
copying mrcnn\utils.py -> build\lib\mrcnn
copying mrcnn\visualize.py -> build\lib\mrcnn
copying mrcnn\__init__.py -> build\lib\mrcnn
creating build\bdist.win-amd64
creating build\bdist.win-amd64\egg
creating build\bdist.win-amd64\egg\mrcnn
copying build\lib\mrcnn\config.py -> build\bdist.win-amd64\egg\mrcnn
copying build\lib\mrcnn\model.py -> build\bdist.win-amd64\egg\mrcnn
```

Gambar 4.5 Proses instalasi Mask R-CNN

c) Instalasi Pycocotools

Setelah berhasil melakukan instalasi Mask-RCNN, selanjutnya adalah instalasi *library* pycocotools. *Library* ini sendiri merupakan salah satu *library* Python yang menyediakan antarmuka untuk bekerja dengan kumpulan *dataset* dari COCO. Pycocotools sangat penting untuk memuat dan mengurai anotasi pada *dataset* COCO, mengevaluasi kinerja model yang sesuai dengan standar *dataset* COCO, serta menangani *segmentation mask* yang sangat berguna untuk menyimpan dan memanipulasi data segmentasi. Untuk melakukan instalasi Pycocotools, langkah pertama adalah masuk kedalam folder cocoapi-master melalui *command prompt*. Contohnya pada komputer penulis, folder tersebut berada pada (envb) `D:\TA\sparkir\cocoapi-master\PythonAPI>`

Di dalam *folder* ini terdapat *file* setup.py yang merupakan *file* untuk melakukan instalasi *library* Pycocotools. Instalasi *library* ini menggunakan perintah python setup.py install yang sama seperti perintah instalasi Mask-RCNN.

```
Installing collected packages: pycocotools
  Running setup.py install for pycocotools ... done
Successfully installed pycocotools-2.0
```

Gambar 4.6 Proses instalasi pycocotools

4.1.2 Training

Sebelum melakukan training, format anotasi dataset yang digunakan untuk melatih model Mask-RCNN pada penelitian ini menggunakan format COCO. Anotasi pada dataset coco memiliki 80 kelas dengan satu background kelas. Dataset yang akan digunakan pada penelitian ini adalah car Image Dataset yang anotasinya sesuai dengan format COCO namun hanya memiliki satu kelas yaitu car. Gambar 4.7 merupakan kategori pada anotasi pada *car Image Dataset.*

```
"categories":[{"id":1,"name":"car","supercategory":"none"},
```

Gambar 4.7 Anotasi pada *car Image Dataset*

Source code untuk mempermudah melakukan *training* model Mask-RCNN telah tersedia pada *repository* github matterport/Mask_RCNN. Pada folder samples/coco terdapat file coco.py yang merupakan konfigurasi *default* untuk melatih model Mask R-CNN. Untuk melatih model, pertama-tama tentukan nilai parameter yang sesuai dengan spesifikasi komputer. Gambar 4.8 merupakan konfigurasi *training* yang digunakan pada penelitian ini.

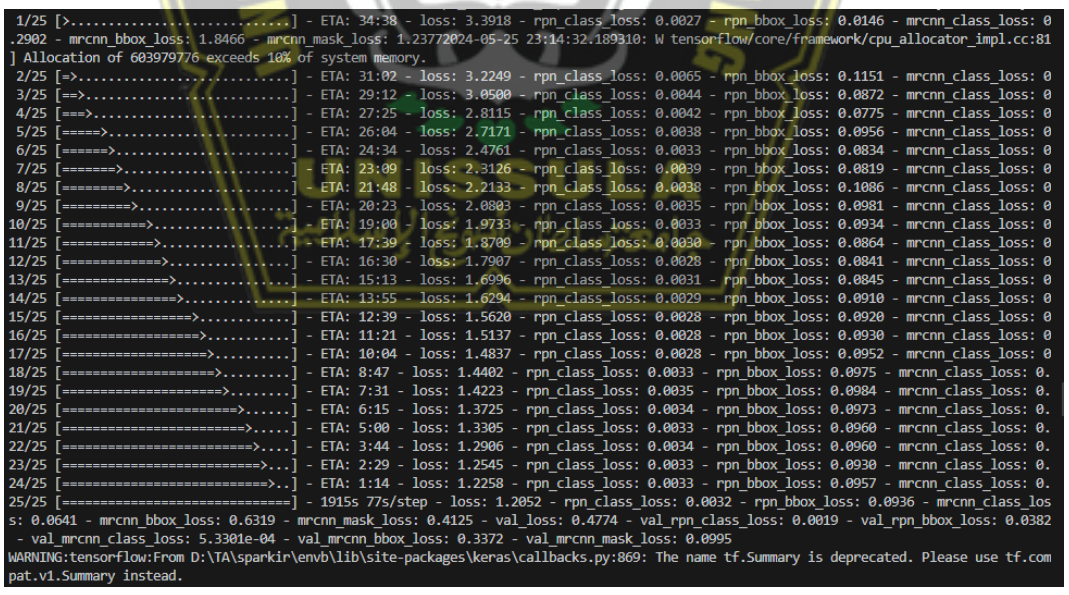
```
class MobilConfig(Config):
    NAME = "mobil_cfg_coco"
    NUM_CLASSES = 1 + 1
    STEPS_PER_EPOCH = 25
    DETECTION_MIN_CONFIDENCE = 0.9
    VALIDATION_STEPS = 2
    LEARNING_RATE = 0.001
    BACKBONE = "resnet101"
    BATCH_SIZE = 32

model.train(dataset_train, dataset_train,
            learning_rate=config.LEARNING_RATE, epochs=10, layers='heads')
```

Gambar 4.8 Konfigurasi *training*

- `NUM_CLASSES`, merupakan jumlah kelas yang akan dikenali oleh model. Disini terdapat dua buah kelas yaitu BG (*Background*) dan *car*.
- `BATCH_SIZE`, adalah jumlah sampel gambar yang akan diproses bersamaan dalam satu waktu melewati jaringan saat proses *training*.
- `STEP_PER_EPOCH`, yaitu jumlah *batch iterations* yang dijalankan dalam satu *epoch*. Perhitungan `STEP_PER_EPOCH` merupakan hasil dari jumlah gambar untuk *training* (805 gambar) dibagi jumlah `BATCH_SIZE`.
- `VALIDATION_STEPS`, merupakan jumlah *batch iterations* yang dijalankan selama proses validasi pada setiap *epoch*. Perhitungan `VALIDATION_STEPS` diambil dari jumlah gambar untuk *training* dibagi `BATCH_SIZE`.
- `DETECTION_MIN_CONFIDENCE`, adalah ambang batas skor keyakinan minimum agar objek yang terdeteksi dianggap valid. Parameter ini ditambahkan untuk melewatkan proses deteksi dengan skor *comfidence* dibawah 0,9.

- `LEARNING_RATE`, yaitu ukuran langkah pada setiap iterasi saat bergerak menuju *loss function* paling minimal. Nilai 0,001 merupakan konfigurasi yang umum digunakan pada banyak model pembelajaran. Dikarenakan rendahnya spesifikasi komputer yang digunakan untuk melakukan *training*, maka nilai `LEARNING_RATE` juga ditetapkan serendah mungkin agar proses *training* dapat berjalan lebih cepat.
- `BACKBONE` digunakan untuk menentukan arsitektur tulang punggung yang dipakai. ResNet-101 dipakai karena efektivitasnya dalam menangkap beragam fitur dari gambar serta memiliki keseimbangan yang baik antara kedalaman dan efisiensi komputasi.

Model dilatih dengan jumlah sepuluh *epoch*. Proses pelatihan pada *epoch* pertama menghabiskan waktu dengan estimasi sebesar 34 menit 38 detik dan untuk menyelesaikan 10 *epoch* membutuhkan waktu kurang lebih 5 jam. Berikut pada gambar 4.9 merupakan proses *training* pada *epoch* pertama dimana untuk setiap satu *epoch* harus melewati 25 *step per epoch*.



Gambar 4.9 Proses training

4.2 Pengujian Blackbox

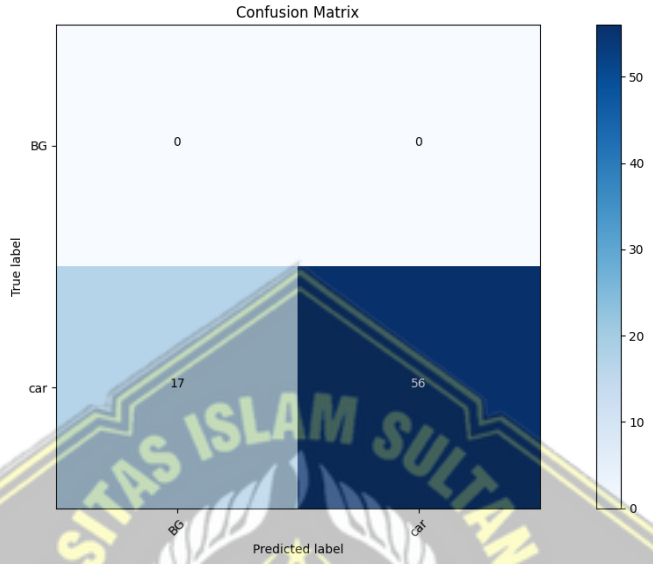
Setelah berhasil melakukan implementasi sistem maka langkah selanjutnya adalah melakukan pengujian dan pengujian yang dipakai pada penelitian ini adalah *blackbox* testing. Pengujian ini bermanfaat untuk melihat apakah perencanaan dan perancangan sistem sudah berfungsi dengan baik atau tidak saat dijalankan melalui pengujian secara fungsionalitas. Tabel 4.1 merupakan tabel untuk pengujian fungsional blackbox.

Tabel 4.1 Tabel pengujian blackbox

No	Skenario Pengujian	Ekspetasi Hasil	Hasil Pengujian	Kesimpulan
1	Melakukan prediksi gambar/video tempat parkir tanpa mobil	Tidak ada yang terdeteksi	Tidak ada yang terdeteksi	Terpenuhi
2	Melakukan prediksi video tempat parkir menggunakan webcam	Webcam terhubung dan dapat mendeteksi mobil	Webcam terhubung dan dapat mendeteksi mobil	Terpenuhi
3	Melakukan Prediksi tempat parkir kosong	Terprediksi kosong	Terprediksi kosong	Terpenuhi
4	Melakukan Prediksi tempat parkir terisi	Terprediksi terisi	Terprediksi terisi	Terpenuhi

4.3 Hasil Evaluasi

Berikut ini hasil evaluasi model Mask R-CNN yang menggunakan data validasi berjumlah 72 gambar mobil.



Gambar 4.10 *Confusion matrix* dari model Mask R-CNN

Gambar 4.10 merupakan *Confusion Matrix* dari model yang telah dibuat. Dari gambar tersebut didapatkan nilai :

- *True Positive* (kanan bawah): 56

  Model tersebut dengan tepat memprediksi "mobil" untuk 56 kejadian yang memang merupakan "mobil".

- *True Negative* (kiri atas): 0

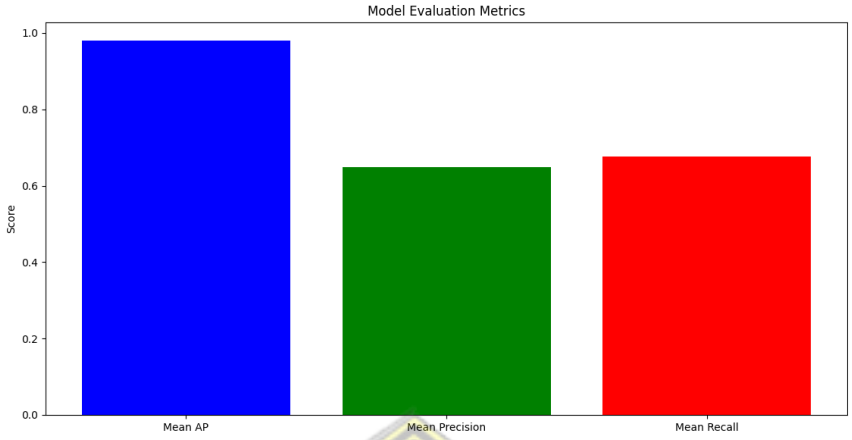
  Model memprediksi dengan tepat "BG" (latar belakang) untuk 0 kejadian yang memang merupakan "BG". Karena tidak ada contoh "BG" pada dataset yang digunakan, hal ini memang wajar terjadi.

- *False Positive* (kanan ats): 0

  Model tersebut salah memprediksi "mobil" untuk 0 kejadian yang sebenarnya adalah "BG".

- *False Negative* (kiri bawah): 17

  Model tersebut gagal memprediksi "mobil" untuk 17 kejadian yang sebenarnya adalah "mobil".



Gambar 4.11 Hasil dari mAP, *Precision*, dan *Recall*

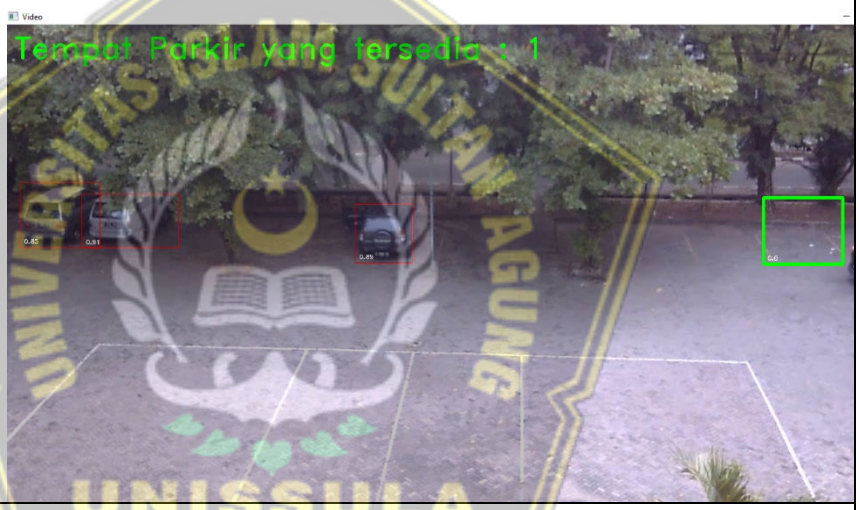
Gambar 4.11 merupakan grafik dari skor rata-rata *Average Precision* (AP), *Precision*, dan *Recall*. Hasil dari skor rata-rata AP berjumlah 0,979 yang menunjukkan bahwa model ini memiliki peforma yang sangat baik dalam hal pertukaran presisi-*recall* serta sangat efektif dalam mengidentifikasi objek dengan benar dan meminimalisir deteksi palsu di berbagai ambang batas IoU. Untuk skor rata-rata presisi berjumlah 0,647 menunjukkan bahwa model tersebut memiliki tingkat *false positive* yang relatif rendah yang mengindikasikan bahwa sebagian besar objek yang terdeteksi sudah benar. Meskipun skor tersebut sudah layak namun masih ada ruang untuk perbaikan untuk mengurangi jumlah *false positive* lebih lanjut. Tidak jauh berbeda skor sebelumnya, skor rata-rata *recall* berjumlah 0,675. Skor ini menunjukkan bahwa model kehilangan sebagian besar bagian dari objek sebenarnya. Meningkatkan *recall* sangatlah penting untuk memastikan lebih banyak bagian dari objek asli yang terdeteksi.

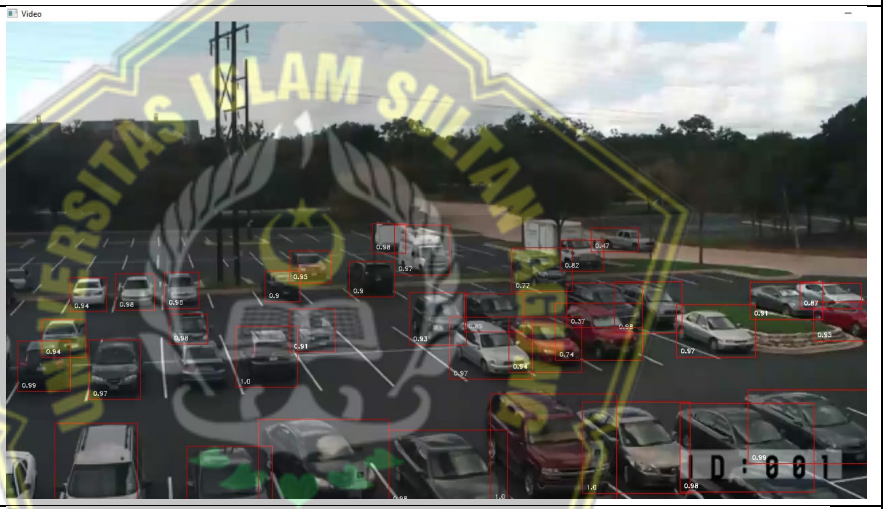
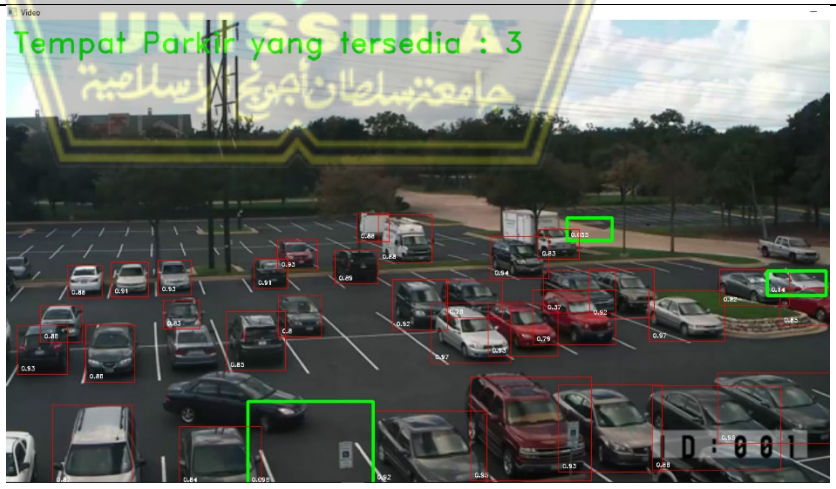
4.4 Hasil Uji Coba Sistem

Berikut ini merupakan hasil pengujian dengan menggunakan empat video yaitu dua video tempat parkir FTI UNISSULA pada sore hari, satu video di pagi hari, dan satu video tempat parkir pagi hari yang diambil dari internet. Sampel video yang diambil dari *webcam* memiliki resolusi 720p. Tabel 4.2 halaman 37 merupakan hasil dari pengujian sistem.

Tabel 4.2 Tabel hasil uji coba sistem

Pengujian	Kasus Pengujian	Hasil
Video tempat parkir FTI UNISSULA pada sore hari 1	Deteksi objek mobil	
	Deteksi ketersediaan tempat parkir	

Video tempat parkir FTI UNISSULA pada sore hari 2	Deteksi objek mobil	
	Deteksi ketersediaan tempat parkir	
Video tempat parkir FTI UNISSULA pada pagi hari	Deteksi objek mobil	

	Deteksi ketersediaan tempat parkir	
Video tempat parkir pada pagi hari (diambil dari internet)	Deteksi objek mobil	
	Deteksi ketersediaan tempat parkir	

Tabel 4.3 Hasil *Confusion Matrix* dari uji coba sistem

Pengujian	TP	TN	FP	FN
Video tempat parkir FTI UNISSULA pada sore hari 1	4	1	0	0
Video tempat parkir FTI UNISSULA pada sore hari 2	3	1	0	0
Video tempat parkir FTI UNISSULA pada pagi hari	3	1	0	1
Video tempat parkir pada pagi hari (diambil dari internet)	34	2	0	1

Tabel 4.3 merupakan tabel hasil *confusion matrix* dari setiap pengujian video 1 hingga 4. Nilai *true* berarti prediksi benar sedangkan *false* salah memprediksi. Nilai *positive* merupakan nilai dari tempat parkir yang terisi, sedangkan nilai *negative* nilai dari tempat parkir yang kosong. Hasil dari TP, TN, FP, dan FN akan dipakai untuk menentukan nilai akurasi, presisi, *recall* dan f-1 *score*. Tidak hanya itu, pada pengujian ini telah dihitung rata-rata *Frame Rate per Second* (FPS) dari masing-masing video pengujian

- $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$
- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1\ score = \frac{2 * precision * recall}{precision+recall}$

1. Evaluasi pengujian pertama
   - $Accuracy = \frac{4+1}{4+0+0+1} = 1$
   - $Precision = \frac{4}{4+0} = 1$
   - $Recall = \frac{4}{4+0} = 1$
   - $F1\ score = \frac{2 * 1 * 1}{1+1} = 1$
   - $FPS = 0{,}35$

2. Evaluasi pengujian kedua
   - *Accuracy*= $\frac{3+1}{3+0+0+1}$ = 1
   - *Precision*= $\frac{3}{3+0}$ = 1
   - *Recall* = $\frac{3}{3+0}$ = 1
   - *F1 score*= $\frac{2*1*1}{1+1}$ = 1
   - *FPS* = 0,35

3. Evaluasi pengujian ketiga
   - *Accuracy*= $\frac{3+1}{3+0+1+1}$ = 0,8
   - *Precision*= $\frac{3}{3+0}$ = 1
   - *Recall* = $\frac{3}{3+1}$ = 0,75
   - *F1 score*= $\frac{2*1*0,75}{1+0,75}$ = 0,857
   - *FPS* = 0,37

4. Evaluasi pengujian keempat
   - Accuracy= $\frac{34+2}{34+0+1+2}$ = 0,972
   - Precision= $\frac{34}{34+0}$ = 1
   - Recall = $\frac{34}{34+1}$ = 0,97
   - F1 score = $\frac{2*1*0,97}{1+0,97}$ = 0,984
   - *FPS* = 0,35

5. Rata-rata pengujian 1 - 4:
   - *Accuracy*= $\frac{1+1+0,8+0,972}{4}$ = 0,943 = 94,3%
   - *Precision*= $\frac{1+1+1+1}{4}$ = 1 = 100%
   - *Recall* = $\frac{1+1+0,75+0,97}{4}$ = 0,93 = 93%
   - *F1 score*= $\frac{1+1+0,857+0,984}{4}$ = 0,96 = 96%
   - *FPS* = 0,36

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Bedasarkan hasil penelitian menunjukkan bahwa deteksi ketersediaan tempat parkir dapat bekerja dengan cukup baik menggunakan laptop dan *webcam*. Berikut adalah beberapa kesimpulan yang bisa diambil dari penelitian ini.

1. Mask R-CNN mampu mendeteksi ketersediaan tempat parkir di Fakultas Teknologi Industri Universitas Islam Sultan Agung dengan hasil yang cukup baik pada laptop karena model telah dilatih menggunakan dataset yang terdiri dari 877 gambar.
2. Pada tahap pengujian sampel didapatkan nilai akurasi sebesar 94,3%, presisi 100%, *recall* 93%, dan skor f1 96%.
3. Jumlah masa (*epoch*), langkah per masa, dan langkah validasi yang diterapkan pada konfigurasi training dapat meningkatkan peforma akurasi dari model yang dilatih.
4. Hasil video pengujian memiliki peforma yang sangat rendah dimana rata-rata *frame rate* video yang dihasilkan adalah 0,36 FPS.

5.2 Saran

Terdapat beberapa saran untuk penelitian ini ataupun untuk penelitian selanjutnya yaitu :

1. Menggunakan perangkat yang lebih mumpuni untuk melatih model maupun menjalankan program dikarenakan kebutuhan spesifikasi untuk menggunakan metode ini cukup tinggi.
2. Menambahkan suatu *user interface* seperti situs web atau aplikasi *mobile*, agar informasi ketersediaan tempat parkir dapat dilihat secara umum.

DAFTAR PUSTAKA

Aa. (2024). *car Dataset*. Roboflow Universe. https://universe.roboflow.com/aa-k6pt5/car-z5zwx

Abdulla, W. (2017). *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. Github. https://github.com/matterport/Mask_RCNN

Assidhiqi, F. (2021). PENGEMBANGAN SISTEM DETEKSI HUNIAN PARKIR MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK. *Tugas Akhir Universitas Islam Indonesia*, *1*(1), 229–237.

Gandhi, R. (2022). A novel approach of object detection using deep learning for animal safety. *International Conference on Cloud Computing, Data Science & Engineering (Confluence)*.

Hanun, N., Sarosa, M., & Andrie Asmara, R. (2023). Pemanfaatan Algoritma Faster R-CNN ResNet-101 Untuk Deteksi Potongan Tubuh Manusia. *Jurnal Elektronika dan Otomasi Industri*, *10*(1), 94–103. https://doi.org/10.33795/elkolind.v10i1.2754

Hozairi. (2021). Implementasi Orange Data Mining Untuk Klasifikasi Kelulusan Mahasiswa Dengan Model K-Nearest Neighbor, Decision Tree Serta Naive Bayes. *Jur nal Ilmiah NERO*, *Vol. 6*, *Is*.

Jupiyandi, S., Saniputra, F. R., Pratama, Y., Dharmawan, M. R., & Cholissodin, I. (2019). Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *6*(4), 413. https://doi.org/10.25126/jtiik.2019641275

Khesya, N. (2021). Mengenal Flowchart Dan Pseudocode Dalam Algoritma Dan Pemrograman. *Osf.Io*.

Limbong, R. (2020). Implementasi Algoritma Unary Coding Pada Kompresi Citra Ultrasonografi. *Pelita Informatika: Informasi dan Informatika*, 367–370.

Lin, Tsung-Yi; Doll´, Piotr; Girshick, Ross; He, Kaiming; Hariharan, Bharath; Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Maulana, R., Fitriyah, H., & Prakasa, E. (2018). Implementasi Sistem Deteksi Slot Parkir Mobil Menggunakan Metode Morfologi dan Background Subtraction. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *2*(5), 1954–1959.

Odemakinde, E. (2021). *Everything about Mask R-CNN: A Beginner's Guide*. viso.ai. https://viso.ai/deep-learning/mask-r-cnn

Putra, A. R., & Candradewi, I. (2017). Deteksi Ketersediaan Slot Parkir Berbasis Pengolahan Citra Digital Menggunakan Metode Histogram of Oriented Gradients dan Support Vector Machine. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, *7*(1), 13. https://doi.org/10.22146/ijeis.15411

Rahman, S., & Dafitri, H. (2019). Aplikasi Simulasi Deteksi Lokasi Parkir Kosong Menggunakan Ektraksi Ciri Objek. *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, *4*(1), 99–104. https://doi.org/10.30743/infotekjar.v4i1.1327

Statistik, B. P. (2020). *Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit), 2018-2020*. https://www.bps.go.id/indicator/17/57/1/jumlah-kendaraan-bermotor.html

Yusnita, R., Fariza, N., & Norazwinawati, B. (2012). Intelligent Parking Space Detection System Based on Image Processing. *International Journal of Innovation, Management and Technology*, *3*(3), 232–235. http://www.ijimt.org/papers/228-G0038.pdf