

BAB II

TINJAUAN PUSTAKA

2.1 Computer Vision

Computer Vision adalah bidang yang mempelajari bagaimana computer mampu melihat dan menangkap informasi yang ada dalam suatu gambar atau video. Berbeda dengan Pengolahan Citra yang memproses suatu gambar menjadi gambar lain untuk memperoleh suatu informasi, sedangkan *computer vision* mempelajari bagaimana komputer dapat menangkap, memproses, menganalisa dan memahami sebuah gambar atau ciri dari suatu citra (Forsyth & Ponce, 2003).

Tujuan dari *computer vision* adalah menghasilkan suatu keputusan dari objek yang nyata secara fisik dan suatu gambaran pengindraan dari suatu citra. Beberapa ahli berpendapat, tujuan dasar dari *computer vision* adalah membangun sebuah deskripsi dari sebuah citra. Karena untuk menghasilkan suatu keputusan dari objek nyata adalah hal yang penting untuk membangun suatu deskripsi dari citra yang ditangkap (Shapiro & Stockman, 2001).

OpenCV (Open Computer Vision) merupakan sebuah *library* perangkat lunak *computer vision* dan *machine learning*. *OpenCV* dapat diunduh secara bebas dan tersedia untuk bahasa pemrograman C, C++, Java dan Python. *OpenCV* berlisensi BSD (Berkeley Software Distribution), sehingga memudahkan bagi pelaku bisnis untuk memanfaatkan dan mengubah kode. *OpenCV* memiliki lebih dari 2500 algoritma optimal, yang mencakup sekumpulan algoritma *computer vision* dan *machine learning*

bertipe klasik dan canggih. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak gerakan kamera, melacak objek yang bergerak, dan lain-lain (Gracia & Suarez, 2015).

2.2 Deteksi Objek

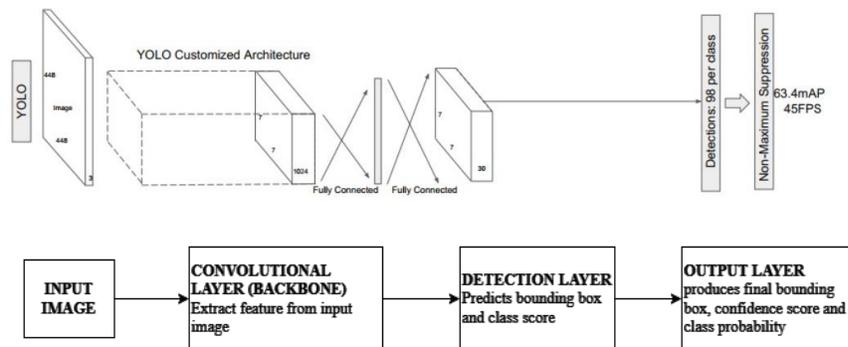
Deteksi objek pada deep learning adalah proses identifikasi dan klasifikasi objek dalam gambar atau video menggunakan model jaringan saraf tiruan yang kompleks. Model-model ini mampu mendeteksi berbagai objek, menentukan lokasinya dengan presisi menggunakan *bounding box*, dan mengklasifikasikan jenis objek tersebut. Deteksi objek menjadi salah satu tugas penting dalam *Computer Vision* yang melibatkan pemahaman dan interpretasi citra digital (Liu, et al., 2016).

2.3 Algoritma YOLO

Algoritma YOLO (*You Only Look Once*) pertama kali diperkenalkan oleh Joseph Redmon merupakan algoritma yang menggunakan jaringan saraf untuk menyediakan deteksi objek secara *real time* (waktu nyata). Algoritma ini digunakan pada berbagai aplikasi untuk mendeteksi sinyal lalu lintas, orang, mobil, tempat parkir dan juga hewan. Deteksi objek di YOLO dilakukan sebagai masalah regresi dan memberikan probabilitas kelas dari gambar yang terdeteksi. Algoritma ini hanya membutuhkan satu propagasi maju melalui jaringan syaraf untuk mendeteksi objek. Ini berarti bahwa prediksi di seluruh gambar dilakukan dalam satu kali algoritma. Digunakan untuk

memprediksi berbagai probabilitas kelas dan kotak pembatas secara bersamaan (Rofii, Priyandoko, Fanani, & Suraji, 2021).

Algoritma YOLO dapat diterapkan dalam bidang seperti *Self-Driving Car* untuk menghindari tabrakan dengan objek sekitar. YOLO juga digunakan mendeteksi berbagai jenis hewan di hutan, beberapa jenis hewan yang dapat dideteksi antara lain : beruang, gajah dan jerapah. YOLO mempunyai kemampuan belajar yang sangat baik yang memungkinkannya mempelajari representasi objek dan menerapkannya dalam deteksi objek (Alfarisi, 2020).



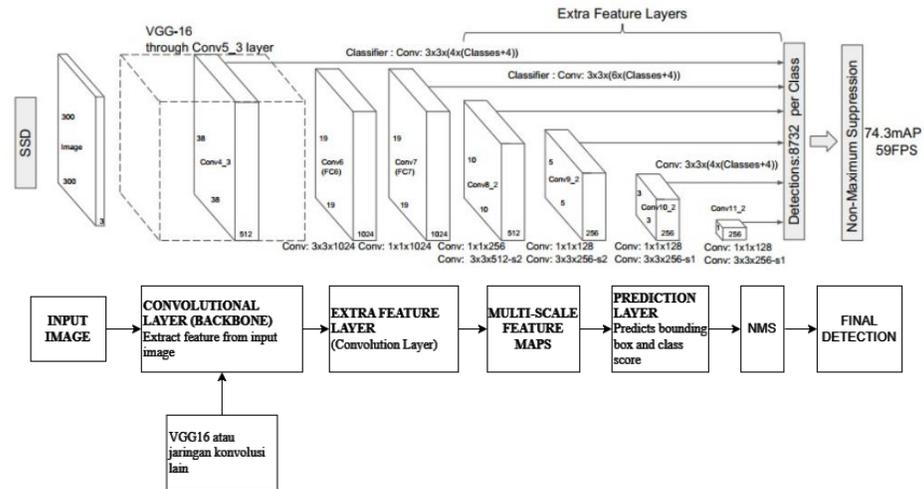
Gambar 2.1 Arsitektur YOLO (sumber : Rančić, K, dkk. (2023). Animal Detection and Counting from UAV Images Using Convolutional Neural Networks. Drones, 7, 179. <https://doi.org/10.3390/drones7030179>)

Arsitektur YOLO terdiri dari beberapa lapisan konvolusional yang diatur untuk memproses input dan menghasilkan prediksi dalam satu langkah. Pada *convolutional layer* ekstrak fitur menggunakan darknet 53 sebagai *backbone*. Setelah fitur diekstraksi oleh *backbone*, fitur tersebut di proses oleh *detection layer* untuk menghasilkan

bounding box. Lalu pada *output layer* menghasilkan *final bounding box*, *confidence score* dan probabilitas kelas (Redmon & Farhadi, 2019).

2.4 Algoritma SSD

Algoritma SSD (*Single Shot MultiBox Detector*) merupakan salah satu metode deteksi objek yang efisien dan cepat. SSD bekerja dengan menggunakan jaringan konvolusi (*Convolutional Neural Network*) yang menghasilkan sejumlah kotak pembatas (*bounding boxes*) dengan ukuran tetap serta skor untuk kehadiran objek kelas tertentu dalam kotak-kotak tersebut. Proses ini diikuti oleh langkah *non-maximum suppression* untuk menghasilkan deteksi akhir. SSD menggabungkan beberapa fitur unik yang membuatnya unggul, seperti penggunaan kotak default (*default boxes*) yang diprediksi pada beberapa peta fitur (*feature maps*) dengan skala yang berbeda. Ini memungkinkan SSD untuk mendeteksi objek pada berbagai skala dan aspek rasio, sehingga meningkatkan akurasi dan kecepatan deteksi. Selain itu, algoritma ini menggunakan filter konvolusi kecil untuk memprediksi skor kategori dan *offset* kotak pembatas, yang disematkan pada peta fitur yang berbeda di jaringan (Liu, et al., 2016).

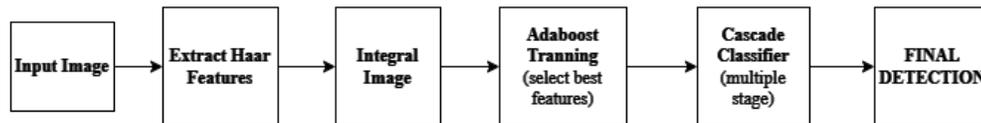


Gambar 2.2 Arsitektur SSD (sumber : Rančić, K, dkk. (2023). Animal Detection and Counting from UAV Images Using Convolutional Neural Networks. *Drones*, 7, 179. <https://doi.org/10.3390/drones7030179>)

Arsitektur SSD terdiri dari beberapa tahapan utama. Pertama, gambar input diproses oleh jaringan dasar (*base network*) seperti VGG16, yang mengekstrak fitur dari gambar dengan melakukan konvolusi dan *pooling* bertingkat. Setelah fitur dasar diekstraksi, SSD menambahkan beberapa lapisan konvolusi tambahan yang menghasilkan peta fitur dengan resolusi berbeda. Setiap peta fitur ini digunakan untuk membuat prediksi *bounding box* dan skor keyakinan (*confidence scores*) untuk setiap kelas objek. Setelah semua prediksi dihasilkan, SSD menggunakan teknik *Non-Maximum Suppression* (NMS) untuk mengeliminasi kotak pembatas yang tumpang tindih (Cao, Kong, Zhang, Li, & Xie, 2020).

2.5 Algoritma Haar Cascade

Algoritma Haar Cascade merupakan salah satu model *Machine Learning* yang kerap kali digunakan sebagai pondasi pada aplikasi *Object Detection* terutama untuk mendeteksi wajah dalam sebuah gambar atau video (Wisesa, 2019). Algoritma ini merupakan gagasan dari Paul Viola dan Michael Jones yang tertuang dalam paper berjudul “*Rapid Object Detection Using A Boosted Cascade of Simple Features*” di tahun 2001. Algoritma ini juga disebut sebagai algoritma Viola Jones.



Gambar 2.3 Arsitektur Haar Cascade

Arsitektur Haar Cascade menerapkan *Cascade Function* dan menggunakan 4 tahap utama (Alfian, Raharjo, & Ibrahim, 2022)

1. Haar Like Feature

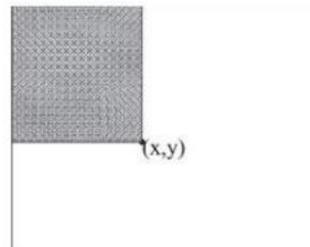
Haar menunjukan istilah matematika (*Haar Wavelet*) yang berbentuk kotak, prinsipnya sama seperti fungsi *fourier* yang awalnya pengolahan gambar hanya dengan melihat nilai RGB setiap piksel, namun metode ini ternyata tidak efektif. Lalu Viola dan Jones mengembangkan *Haar Like Feature* yang memproses gambar dalam kotak-kotak dimana dalam satu kotak terdapat beberapa piksel. Setiap kotak kemudian diproses dan menghasilkan perbedaan nilai yang menandakan daerah terang dan daerah gelap. Selisih jumlah nilai

piksel daerah terang dengan jumlah nilai piksel daerah gelap disebut dengan nilai *Haar Like Feature*.

$$F_{Haar} = \sum F_{white} - \sum F_{black}$$

2. Integral Image

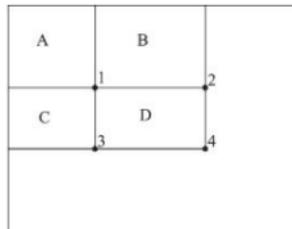
Integral image adalah suatu teknik untuk menghitung nilai fitur secara cepat dengan mengubah nilai setiap piksel menjadi suatu representasi citra baru.



Gambar 2.4 Integral Image Haar Cascade

3. Adaboost Learning

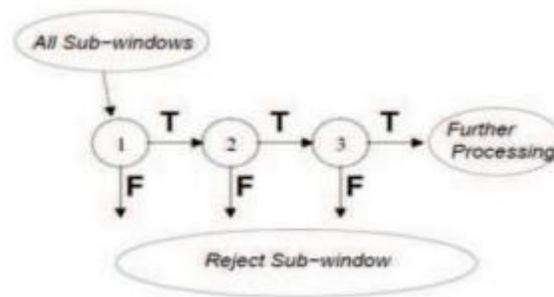
Perhitungan nilai dari suatu fitur dapat dilakukan secara cepat dengan menghitung nilai citra integral pada empat buah titik.



Gambar 2.5 Adabost Learning Haar Cascde

4. Cascade Classifier

Pada tahap ini dilakukan dengan mengkombinasikan *classifier* yang kompleks dalam sebuah struktur bertingkat yang dapat meningkatkan kecepatan pendeteksian objek dengan memfokuskan pada daerah citra yang berpeluang saja. Adapun alur kerja dari *Cascade Classifier* sebagai berikut:



Gambar 2.6 Haar Cascade Classifier

Tahap 1 : setiap sub citra akan diklasifikasikan dengan satu fitur, bila hasil tidak memenuhi kriteria maka hasil ditolak

Tahap 2 : setiap sub citra kembali diklasifikasikan, jika didapat nilai *threshold* yang diinginkan maka dilanjutkan ke tahap 3

Tahap 3 : sub citra yang lolos akan berkurang hingga mendekati citra yang ada pada sampel

2.5 State Of The Art

Tabel 2.1 SOTA

| No | Judul | Penulis dan Tahun | Metode / solusi | Hasil penelitian |
|----|--|---|--|--|
| 1 | Implementasi Algoritma CNN dan YOLO untuk Mendeteksi Jenis Kendaraan pada Jalan Raya | Fernandy Jupiter, Edi Surya Negara, Yesi Novaria Kunang dan M. Izman Herdiansyah (2023) | YOLO (<i>You Only Look Once</i>) dan CNN (<i>Convolutional Neural Network</i>) | Hasil penelitian menunjukkan bahwa pendekatan gabungan CNN dan YOLO menghasilkan tingkat akurasi yang kurang baik dalam mendeteksi jenis kendaraan pada jalan raya yang ditunjukkan dengan nilai presisi sebesar 0,50, nilai <i>recall</i> sebesar 0,48 dan nilai <i>f1-score</i> sebesar 0,46 pada epoch 50. Namun pada saat pendeteksian pada masing-masing algoritma menghasilkan tingkat akurasi yang baik, hal ini ditunjukkan dengan nilai pada saat |

| | | | | |
|---|---|---|---|--|
| | | | | <p>pedeteksian pada algoritma YOLO yakni pada parameter presisi sebesar 0,936, nilai <i>recall</i> sebesar 0,914 dan nilai <i>f1-score</i> sebesar 0,924, kemudian pada algoritma CNN yang ditunjukkan pada nilai presisi sebesar 0,95, nilai <i>recall</i> sebesar 0,95 dan <i>f1-score</i> sebesar 0,95.</p> |
| 2 | <p>Metode CNN Dan Metode Haar Cascade Untuk Mendeteksi Sepeda Motor Yang Melintasi Area Trotoar</p> | <p>Utsman Hadi Zaini Dan Abd.Rabi' (2023)</p> | <p>CNN (<i>Convolutional Neural Network</i>) dan Haar Cascade</p> | <p>Hasil penelitian tersebut didapatkan perbandingan akurasi yang cukup jauh, metode CNN mendapatkan hasil akurasi sebesar 88,4% dan metode Haar Cascade mendapatkan hasil akurasi 63,4%, dari hasil pengujian yang diperoleh metode CNN lebih unggul akurasinya daripada haar cascade.</p> |

| | | | | |
|---|---|---|---------------------------------------|--|
| 3 | Implementasi Sistem Penghitung Kendaraan Otomatis Berbasis <i>Computer Vision</i> | Dolly Indra, Herman dan Firman Shantya Budi (2023) | SSD (Single Shot Multibox Detector) | Penelitian tersebut dilakukan untuk membuat Sistem Otomatis Penghitung Kendaraan yang diimplementasikan pada jasa steam (cuci mobil dan motor). Metode pendeteksian kendaraan dengan menggunakan SSDMobilenetV2 sudah mampu mendeteksi kendaraan mobil dan motor dengan rata rata tingkat akurasi sebesar 46,6%. |
| 4 | Perbandingan Metode Haar Cascade, YOLO V3 dan TinyYOLO V3 Dalam Mendeteksi | Marzuarman, Stephan, Muharnis, Azizul, Doni Mirza Rinaldi dan Bagas | Haar Cascade, YOLO V3 dan TinyYOLO V3 | Pada Metode YOLO V3 menghasilkan akurasi yang baik namun kecepatan pergerakan lambat akibat proses <i>learning</i> data yang lama sehingga menambah beban kerja dari komputer yang |

| | | | | |
|---|---|---|---------------------------|--|
| | Kendaraan Bermotor Berbasis Video | Prasetyo (2022) | | digunakan. Pada metode Haar Cascade menghasilkan kecepatan tampilan yang baik namun tingkat akurasi deteksi masih kurang baik dibandingkan YOLO V3. Sedangkan pada TinyYOLO V3 menghasilkan kecepatan tampilan gambar yang lebih baik dari YOLO V3 namun tingkat akurasi masih kalah dibanding kedua algoritma sebelumnya. |
| 5 | <i>Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs</i> | Upesh Nepal dan Hossein Eslamiat (2022) | YOLOv3, YOLOv4 and YOLOv5 | Akurasi algoritma YOLO V5 lebih baik dibandingkan YOLO V3 dan YOLO V4 sedangkan untuk kecepatan waktu eksekusi YOLO V3 memiliki kecepatan yang lebih baik dibandingkan YOLO V4 dan YOLO V5. |

| | | | | |
|---|--|--|--------------------------------|--|
| 6 | <p>Perancangan Sistem Pendeteksi Kepadatan Lalu Lintas Menggunakan Metode <i>Haar Cascade Classifier</i></p> | <p>Muhammad Phasya Alfian, Jangkung Raharjo dan Nur Ibrahim (2022)</p> | <p>Haar Cascade Classifier</p> | <p>Nilai uji data selama 5 menit pada pagi hari menghasilkan akurasi data sebesar 88,02%, pada siang hari menghasilkan akurasi data sebesar 63,04 % dan pada sore hari menghasilkan akurasi data sebesar 75,98%. Berdasarkan hasil tersebut nilai akurasi tertinggi dilakukan pada pagi hari karena intensitas cahaya yang tidak tinggi sehingga kendaraan yang melintas dapat dideteksi dengan baik sedangkan nilai akurasi terendah adalah siang hari saat intensitas cahaya paling tinggi yang menyebabkan kamera menjadi silau dan objek yang dideteksi menjadi tidak jelas.</p> |
|---|--|--|--------------------------------|--|

| | | | | |
|---|--|--|-------------------------------|--|
| 7 | Eksperimen Penerapan Sistem <i>Traffic Counting</i> dengan Algoritma YOLO (You Only Look Once) V.4. | Yovi Pratama dan Errissya Rasywir (2021) | YOLO (You Only Look Once) V.4 | Dari hasil eksperimen yang dilakukan dalam penelitian ini, diperoleh hasil deteksi yang mempunyai akurasi yang cukup baik pada hasil pemmisahan frame-frame dari video data. Transformasi posisi, dimensi, komposisi dan direksi yang tidak beraturan tetap dapat ditangkap sebagai fitur yang sama. |
| 8 | Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir berbasis Python dan YOLO v4 | Getsa Novandra Rizkatama, Anan Nugroho dan Alfa Faridh Suni (2021) | YOLO (You Only Look Once) V.4 | Pendeteksi jumlah mobil memperoleh akurasi total sebesar 72.8% sehingga program tersebut dapat digunakan untuk menghitung ketersediaan lahan parker. |

| | | | | |
|---|--|--|--------------------------------------|---|
| 9 | <p>Peningkatan Akurasi Penghitungan Jumlah Kendaraan dengan Membangkitkan Urutan Identitas Deteksi Berbasis Yolov4 <i>Deep Neural Networks</i></p> | <p>Faqih Rofii, Gigih Priyandoko, M. Ifan Fanani dan Aji Suraji (2021)</p> | <p>YOLO (You Only Look Once) V.4</p> | <p>Penelitian ini telah mengembangkan penggunaan detektor objek berbasis Yolov4 untuk mendeteksi objek kendaraan. Berdasarkan hasil pengujian kamera 1 dengan ketinggian 6 meter di atas permukaan jalan dengan sudut 50 derajat terhadap tanah pada saat malam hari didapat nilai akurasi total, presisi total dan recall total masing-masing adalah sebesar 68%, 83% dan 85% sedangkan pada saat siang hari dengan kamera 1 nilai akurasi total, presisi total dan <i>recall</i> total masing-masing adalah sebesar 83%, 93% dan 94%. hasil pengujian kamera 1 dengan ketinggian 1,5 meter di atas permukaan jalan dengan</p> |
|---|--|--|--------------------------------------|---|

| | | | | |
|----|--|--|---------------------|---|
| | | | | <p>sudut 90 derajat terhadap tanah pada saat malam hari didapat nilai akurasi total, presisi total dan <i>recall</i> total masing-masing adalah sebesar 68%, 77% dan 78% sedangkan pada saat siang hari dengan kamera 1 nilai akurasi total, presisi total dan <i>recall</i> total masing-masing adalah sebesar 77%, 84% dan 85%.</p> |
| 10 | <p>Pendeteksian Ruang Kosong Parkir di dalam Ruangan</p> | <p>Nunik Afriliana, Rosalina dan Regina Valeria (2018)</p> | <p>Haar Cascade</p> | <p>Hasil penelitian berupa sistem yang dapat mendeteksi mobil di tempat parkir tertentu dan secara otomatis mengenali ruang sebagai deretan slot parker sehingga tempat parker ditemukan dengan tepat. Berdasarkan hasil pengujian, 72,7 % sistem dapat mengenali ruang parkir kosong.</p> |

Penelitian yang akan dilakukan memiliki keterkaitan dengan penelitian-penelitian sebelumnya yaitu mengenai deteksi kendaraan dengan menggunakan Algoritma YOLO (You Only Look Once), SSD (Single Shot Multibox Detector) dan Haar Cascade. Penelitian ini bertujuan untuk mengetahui perbandingan performa dari ketiga algoritma tersebut.

Selain itu penelitian ini juga bertujuan untuk mengevaluasi hasil deteksi dan perhitungan dari algoritma YOLO, SSD dan Haar Cascade. Evaluasi dilakukan dengan membandingkan hasil deteksi ketiga algoritma dengan jumlah sebenarnya. Seperti penelitian yang berjudul “Perbandingan Metode Haar Cascade, YOLO V3 dan TinyYOLO V3 Dalam Mendeteksi Kendaraan Bermotor Berbasis Video”. Penelitian tersebut dilakukan untuk mengetahui nilai akurasi terbaik dari ketiga algoritma tersebut. Hasil dari penelitian menunjukkan metode YOLOV3 mendapatkan tingkat akurasi yang paling baik yaitu sebesar 74%. Untuk metode Haar cascade mendapatkan nilai akurasi sebesar 41%, dan TinyYOLOV3 menghasilkan tingkat akurasi sebesar 25% (Marzuarman, et al., 2022).

Pada penelitian yang berjudul “Implementasi Algoritma CNN dan YOLO untuk Mendeteksi Jenis Kendaraan pada Jalan Raya” mengkombinasikan Algoritma YOLO dan CNN untuk mendeteksi kendaraan di jalan raya. Dengan mengkombinasikan kedua algoritma tersebut mendapatkan hasil yang kurang baik yaitu nilai presisi 0,5, nilai *recall* 0,48 dan *F1-Score* sebesar 0,46. Sedangkan pada saat pendeteksian pada masing-

masing algoritma menghasilkan yang baik, hal ini ditunjukkan dengan nilai pada saat pendeteksian pada algoritma YOLO yakni pada parameter presisi sebesar 0,936, nilai *recall* sebesar 0,914 dan nilai *f1-score* sebesar 0,924, kemudian pada algoritma CNN yang ditunjukkan pada nilai presisi sebesar 0,95, nilai *recall* sebesar 0,95 dan *f1-score* sebesar 0,95 (Jupiter, Negara, Kunang, & Herdiansyah, 2023).

Kemudian pada penelitian yang berjudul “Metode CNN Dan Metode Haar Cascade Untuk Mendeteksi Sepeda Motor Yang Melintasi Area Trotoar” membandingkan performa dari algoritma CNN dan Haar Cascade dalam mendeteksi Sepeda Motor yang melanggar dan memasuki area trotoar . Hasil Deteksi dengan menggunakan metode CNN mendapatkan hasil akurasi sebesar 88,4% dan metode Haar Cascade mendapatkan hasil akurasi 63,4% (Zaini & Abd.Rabi, 2023).

Meskipun penelitian yang menerapkan Algoritma YOLO, SSD dan Haar Cascade telah dilakukan. Sebagian besar penelitian belum meneliti mengenai pengaruh dari perubahan resolusi video dan kondisi pencahayaan pada video. Penelitian ini akan mendeteksi dan menghitung jumlah mobil yang melintas menggunakan tiga algoritma yaitu YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector) dan Haar Cascade. Keterbaruan dari penelitian ini yaitu dengan menganalisis pengaruh perubahan resolusi video dan nilai *Brightness* terhadap jumlah hasil deteksi dan nilai akurasi dari ketiga algoritma tersebut.

2.6 Keterbaruan penelitian

Tabel 2.2 Kebaruan Penelitian

| No | Judul | Penulis Dan Tahun | Ruang LIngkup | | | | | | | | |
|----|--|---|---------------|-----|--------------|----------|--------------|--------------|-------|-----------|--------------|
| | | | Metode | | | Tujuan | | | Objek | | |
| | | | YOLO | SSD | Haar Cascade | Analisis | Perbandingan | Implementasi | Cuaca | Kendaraan | Ruang / Area |
| 1 | Implementasi Algoritma CNN dan YOLO untuk Mendeteksi Jenis Kendaraan pada Jalan Raya | Fernandy Jupiter, Edi Surya Negara, Yesi Novaria Kunang dan M. Izman Herdiansyah (2023) | √ | - | - | √ | √ | - | - | √ | - |
| 2 | Metode CNN Dan Metode Haar Cascade Untuk Mendeteksi Sepeda Motor Yang Melintasi Area Trotoar | Utsman Hadi Zaini Dan Abd.Rabi' (2023) | - | - | √ | - | √ | - | - | √ | - |
| 3 | Implementasi Sistem Penghitung Kendaraan Otomatis Berbasis Computer Vision | Dolly Indra, Herman dan Firman Shantya Budi (2023) | - | √ | - | - | - | √ | - | √ | - |

| | | | | | | | | | | | |
|---|--|---|---|---|---|---|---|---|---|---|---|
| 4 | Perbandingan Metode Haar Cascade, YOLO V3 dan TinyYOLO V3 Dalam Mendeteksi Kendaraan Bermotor Berbasis Video | Marzuarman, Stephan, Muharnis, Azizul, Doni Mirza Rinaldi dan Bagas Prasetyo (2022) | √ | - | √ | - | √ | - | - | √ | - |
| 5 | Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs | Upesh Nepal dan Hossein Eslamiat (2022) | √ | - | - | - | √ | - | - | - | √ |
| 6 | Perancangan Sistem Pendeteksi Kepadatan Lalu Lintas Menggunakan Metode Haar Cascade Classifier | Muhammad Phasya Alfian, Jangkung Raharjo dan Nur Ibrahim (2022) | - | - | √ | - | - | √ | - | √ | - |
| 7 | Eksperimen Penerapan Sistem Traffic Counting dengan Algoritma YOLO (You Only Look Once) V.4. | Yovi Pratama dan Errissya Rasywir (2021) | √ | - | - | - | - | √ | - | √ | - |

| | | | | | | | | | | | |
|----|--|---|---|---|---|---|---|---|---|---|---|
| 8 | Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir berbasis Python dan YOLO v4 | Getsa Novandra Rizkatama, Anan Nugroho dan Alfa Faridh Suni (2021) | √ | - | - | - | - | √ | - | - | √ |
| 9 | Peningkatan Akurasi Penghitungan Jumlah Kendaraan dengan Membangkitkan Urutan Identitas Deteksi Berbasis Yolov4 Deep Neural Networks | Faqih Rofii, Gigih Priyandoko, M. Ifan Fanani dan Aji Suraji (2021) | √ | - | - | √ | - | - | - | √ | - |
| 10 | Implementasi Machine Learning dan Computer Vision pada Pengembangan Sistem Otomasi Klasifikasi dan Perhitungan Kendaraan | Dian Agustiani (2019) | - | √ | - | - | - | √ | - | √ | - |
| 11 | Pendeteksian Ruang Kosong Parkir di dalam Ruangan | Nunik Afriliana, Rosalina dan Regina Valeria (2018) | - | - | √ | - | - | √ | - | - | √ |

| | | | | | | | | | | | | |
|----|--|-----------------------|--|---|---|---|--|---|---|---|---|---|
| 12 | Analisis Perbandingan Algoritma YOLO V.4, SSD dan Haar Cascade Dalam Menghitung Jumlah Mobil Yang Melintas | Dias Ramadhani (2024) | | √ | √ | √ | | √ | - | - | √ | - |
|----|--|-----------------------|--|---|---|---|--|---|---|---|---|---|