

**BAB II**  
**TINJAUAN PUSTAKA**

**2.1 Tinjauan Pustaka**

Penelitian ini didukung oleh tinjauan pustaka yang relevan, yang rinciannya tercantum dalam Tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Hasil
1	(Amwin, 2021)	Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma <i>You Only Look Once</i> (Yolo)	Nilai AP pada mobil sebesar 99,88%, pada sepeda motor 97,79%, becak sebesar 100%, truk sebesar 100%, dan bus 99,09%. Sedangkan nilai mAP sebesar 99,35% dengan waktu pemrosesan selama 4 detik.  Kualitas video dapat mempengaruhi pendeteksian karena semakin tinggi kualitas video maka hasil klasifikasi dan <i>bounding box</i> semakin tinggi pula
2	(Cholid, 2021)	Penerapan Metode <i>You Only Look Once</i> (Yolo) Dan <i>Support Vector Regression</i> (Svr) Untuk Perhitungan Kepadatan Lalu Lintas Berdasarkan <i>Area Occupancy</i>	Metode YOLO dapat mengenali jenis kendaraan dan memperoleh akurasi 75,16% pada kondisi lalu lintas siang hari. Untuk estimasi kepadatan lalu lintas berdasarkan okupansi diterapkan metode YOLO dan SVR. Hal ini direpresentasikan dengan <i>kernel olinomial</i> dengan parameter

Tabel Lanjutan 2. 1 Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Hasil
			optimasi epsilon = 1,0, derajat = 1, gamma = 0,0, dan coef0 = 2,0 memperoleh skor MAPE 53,59; nilai ini lebih kecil dari penggunaan <i>kernel</i> linier mendapatkan nilai MAPE sebesar 55,5
3	(Pratama & Rasywir, 2021)	Eksperimen Penerapan Sistem <i>Traffic Counting</i> dengan Algoritma YOLO ( <i>You Only Look Once</i> ) V.4.	Diperoleh hasil deteksi yang mempunyai akurasi yang cukup baik pada hasil pemisahan <i>frame-frame</i> dari video data. Walaupun tidak seluruh hasil penggunaan algoritma ini sempurna pada semua data, namun didapatkan hasil cenderung baik.
4	(Khatami, 2022)	Deteksi Kendaraan Menggunakan Algoritma <i>You Only Look Once</i> (Yolo) V3	Didapatkan nilai <i>Precision</i> yang sempurna, nilai <i>Recall</i> di atas 70%, dan <i>F1 Score</i> di atas 80%. Dari kedua pengujian bobot jaringan tersebut, diketahui bahwa objek mobil dan kendaraan besar merupakan objek yang paling berhasil dapat dideteksi oleh ketiga jaringan tersebut. Hal ini disebabkan oleh ukuran dan fitur fisik objek mobil dan kendaraan besar yang hampir mirip antara satu sama lain.

Tabel Lanjutan 2. 2 Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Hasil
5	(Isnaini, 2020)	Aplikasi Penghitung Kendaraan yang Melintas di Jalan Raya Berdasarkan Metode <i>Yolo Object Detection</i>	Dengan pengujian menggunakan 3 video yang diambil dari hasil rekam dan dari internet yang sudah bisa mendeteksi dan menghitung dengan akurat yang mempunyai nilai keakuratan paling tinggi 56% dan paling rendah 10%.
6	(Fachrie, 2020)	<i>A Simple Vehicle Counting System Using Deep Learning with YOLOv3 Model</i>	Penghitungan didasarkan pada empat jenis kendaraan yaitu mobil, sepeda motor, bus, dan truk. Hasilnya, sistem yang di usulkan mampu menghitung kendaraan yang melintasi jalan berdasarkan video yang direkam oleh kamera dengan akurasi tertinggi 97,72%.
7	(Sumarudin et al., 2019)	Aplikasi Penghitung Kendaraan Pada Jalur Pantura Menggunakan Blob Deteksi dan Kalman Filter	Dapat disimpulkan penggunaan kalman <i>filter</i> memiliki nilai deteksi 78.81% sehingga dapat diimplementasikan dalam sistem pendeteksian jumlah kendaraan di jalur pantura.
8	(Faisal & Abadi, 2021)	Implementasi SSD_Resnet50_V1 Untuk Penghitung Kendaraan	Aplikasi dengan model deteksi SSD_Resnet50_v1 mendapatkan akurasi sebesar 56,49% untuk kendaraan berjenis motor, dan 54,43% untuk mobil. Hasil akurasi terhitung rendah dikarenakan model deteksi.

Tabel Lanjutan 2. 3 Penelitian Terdahulu

			<p>kurang bisa mengenali objek kendaraan pada malam hari, sehingga banyak kendaraan yang tidak terdeteksi dan ada kendaraan berjenis motor yang dideteksi sebagai mobil</p>
9	(Majumder & Wilmot, 2023)	<p><i>Automated Vehicle Counting from Pre-Recorded Video Using You Only Look Once (YOLO) Object Detection Model</i></p>	<p>Keakuratan penghitungan otomatis dievaluasi dibandingkan dengan penghitungan manual, dan ditemukan sekitar 90 persen. Selain itu, analisis manfaat-biaya (B/C) menunjukkan bahwa menerapkan metode penghitungan otomatis menghasilkan 1,76 kali lipat investasi.</p>
10	(Ramadhan et al., 2023)	<p>Prototype penghitung Jumlah Dan Kecepatan Kendaraan Otomatis Secara <i>Real Time</i> Berbasis <i>Computer Vision</i> Menggunakan Metode <i>Background Subtraction</i></p>	<p>Untuk raspberry pi mendapatkan rata-rata persentasi akurasi ketepatan sebesar 15% sedangkan untuk laptop mendapat rata-rata persentasi akurasi ketepatan sebesar 90%. Proses perhitungan jumlah kendaraan kurang baik dilakukan menggunakan raspberry pi yang disebabkan <i>drop FPS (frame rate per second)</i> yang sangat jauh.</p>

Pada Tabel 2.1 telah di perlihatkan tentang penelitian terdahulu yang sejenis dengan penelitian ini. Namun, penelitian ini memiliki fokus yang berbeda, yaitu pada analisis akurasi pendeteksian dan penghitungan kendaraan menggunakan teknologi *Computer Vision* dengan algoritma YOLOv8 pada CCTV jalan raya di Kota Bandar Lampung. Selain itu, penelitian ini juga menggunakan perangkat laptop dengan *platform Google Colaboratory* untuk melakukan pengujian. Terdapat empat metrik yang akan digunakan untuk mengevaluasi akurasi hasil pengujian, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Oleh karena itu, penelitian ini diharapkan dapat memberikan kontribusi baru pada pengembangan teknologi penghitung kendaraan dengan tingkat akurasi yang lebih baik.

## 2.2 Kendaraan

Kendaraan merupakan alat transportasi yang berguna sebagai pengangkut orang atau barang dari suatu tempat ke tempat lainnya. Kendaraan dapat berupa mobil, motor, truk, bus, kereta api, kapal laut, dan pesawat terbang. Jenis kendaraan yang digunakan tergantung pada tujuan dan jarak tempuh yang ingin dicapai. Kendaraan umumnya terdiri dari beberapa komponen utama seperti mesin, roda, sistem suspensi, sistem pengereman, dan sistem kemudi. Setiap komponen ini memiliki peran penting dalam menghasilkan kinerja yang optimal dari kendaraan. Selain itu, ada juga teknologi terbaru yang digunakan pada kendaraan seperti teknologi *Computer Vision* dan *Machine Learning* untuk meningkatkan efisiensi dan keamanan dalam berkendara.

Seiring berjalannya waktu pertumbuhan jumlah kendaraan terus meningkat. dalam penggunaannya sendiri, kendaraan dapat memberikan berbagai dampak positif dan negatif. Di satu sisi, kendaraan dapat meningkatkan mobilitas dan aksesibilitas ke berbagai tempat. Namun, di sisi lain, penggunaan kendaraan juga dapat menimbulkan masalah seperti kemacetan, polusi udara, dan kecelakaan lalu lintas. Pertumbuhan jumlah kendaraan perlu disikapi secara bijak (Priyambodo, 2018). Oleh karena itu, perlu dilakukan pengaturan dan pengawasan yang baik dalam penggunaan kendaraan untuk meminimalkan dampak negatif dan memaksimalkan manfaatnya bagi masyarakat.

Setiap jenis kendaraan memiliki karakteristik yang berbeda-beda tergantung dari jenis kendaraannya (Khatami, 2022). Salah satu karakteristik kendaraan yang penting adalah dimensinya, yaitu panjang, lebar, dan tinggi kendaraan. Dimensi kendaraan menjadi sangat penting untuk memastikan bahwa kendaraan tersebut dapat melewati jalan atau jembatan yang ada dengan aman dan tanpa mengganggu lalu lintas yang lain. Selain itu, dimensi kendaraan juga memengaruhi kapasitas muatan yang dapat diangkut oleh kendaraan tersebut. Selain dimensi, karakteristik kendaraan lainnya adalah kecepatan maksimum dan akselerasi kendaraan. Kecepatan maksimum kendaraan menunjukkan seberapa cepat kendaraan dapat berjalan, sedangkan akselerasi kendaraan menunjukkan seberapa cepat kendaraan dapat meningkatkan kecepatannya dari keadaan diam.

### **2.3 Jalan Raya**

Jalan raya merupakan sebuah infrastruktur transportasi darat yang biasanya dilalui oleh kendaraan bermotor untuk menghubungkan berbagai wilayah atau daerah. Jalan raya dipergunakan sebagai tempat akumulasi dari berbagai kendaraan bermotor maupun tak bermotor (Wibisana, 2009). Jalan raya umumnya terdiri dari beberapa jalur yang dipisahkan oleh pembatas jalan atau marka jalan. Karakteristik jalan raya bervariasi tergantung pada tipe jalan, lokasi, dan lalu lintas yang melaluinya. Namun secara umum, jalan raya memiliki karakteristik seperti lebar jalan yang memadai, permukaan jalan yang rata dan halus, dan terdapat aturan lalu lintas yang jelas untuk menjaga keselamatan pengguna jalan. Jalan raya juga dapat dilengkapi dengan fasilitas pendukung seperti jembatan, *flyover*, *underpass*, *rest area*, serta CCTV untuk memantau lalu lintas dan keamanan pengguna jalan.

### **2.4 Deep Learning**

*Deep Learning* adalah salah satu teknik pembelajaran mesin (*Machine Learning*) yang memungkinkan mesin atau komputer untuk mempelajari representasi yang lebih abstrak dan kompleks dari data. Teknik ini dilakukan dengan menggunakan arsitektur jaringan saraf tiruan (*neural network*) yang sangat dalam dengan banyak lapisan (*layer*) dan parameter yang sangat besar. Pada *Deep Learning*, data yang diberikan kepada mesin tidak perlu diproses secara manual atau dipilih fiturnya secara terpisah. Sebaliknya, mesin akan mempelajari fitur-fitur

yang paling penting secara otomatis dengan melakukan optimasi pada parameter jaringan saraf tiruan yang terdiri dari ratusan bahkan ribuan *neuron* . Dengan demikian, *Deep Learning* sangat berguna dalam mengolah data yang kompleks seperti gambar, suara, dan teks. Teknik ini telah banyak digunakan dalam berbagai bidang seperti visi komputer, pengenalan suara, pengenalan tulisan tangan, dan bahkan prediksi kejadian masa depan.

*Deep Learning* dapat didefinisikan sebagai salah satu kelompok algoritma dalam *Machine Learning* yang menggunakan beberapa lapisan pemrosesan nonlinier yang disusun secara berturut-turut untuk melakukan ekstraksi fitur dan transformasi data (Diponegoro et al., 2021). Dalam *Deep Learning*, terdapat banyak lapisan yang berbeda dalam memproses data secara bertahap untuk mencapai tujuan tertentu. Setiap lapisan tersebut akan mempelajari representasi data yang berbeda dan berguna untuk mengekstraksi fitur yang diperlukan dalam memahami pola data yang rumit. Oleh karena itu, DL sering digunakan dalam aplikasi yang membutuhkan pemrosesan data yang kompleks, seperti pengenalan suara, pengenalan wajah, pengenalan objek, dan sebagainya.

*Deep Learning* memiliki banyak domain aplikasi yang berkembang pesat, termasuk diantaranya adalah *Computer Vision* (Shorten et al., 2021). Penerapan *Deep Learning* pada *Computer Vision* memungkinkan komputer untuk memproses dan memahami gambar dan video dengan cara yang sama seperti manusia, termasuk melakukan tugas-tugas seperti deteksi objek, pengenalan wajah, dan klasifikasi gambar secara otomatis dan akurat. Oleh karena itu, *Deep Learning* dan *Computer Vision* adalah bidang yang sangat menjanjikan untuk dijelajahi dan dikembangkan lebih lanjut di masa depan.

## **2.5 *Image Processing***

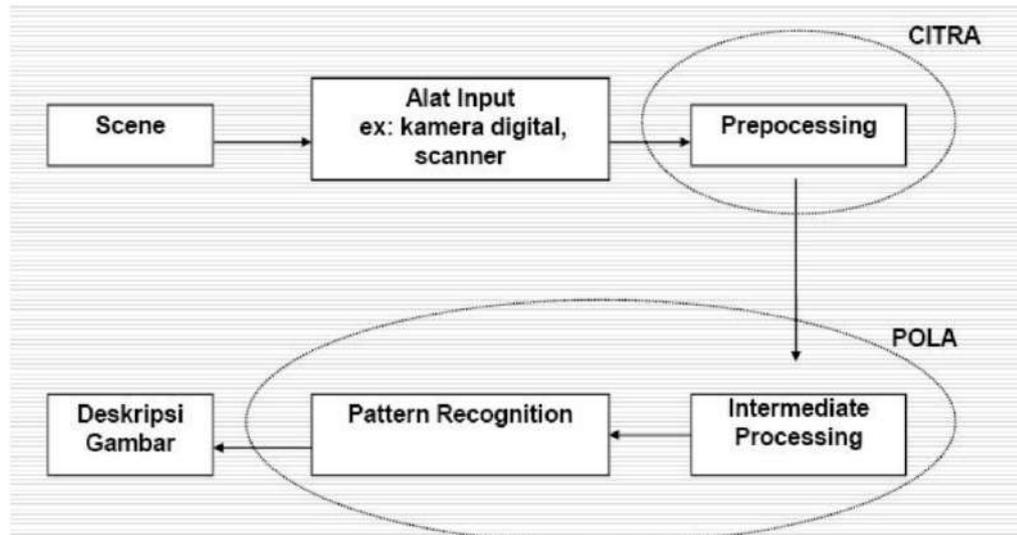
*Image Processing* adalah suatu proses pengolahan citra atau gambar untuk mendapatkan informasi yang berguna dari gambar tersebut. Proses ini dapat dilakukan dengan menggunakan teknik-teknik seperti *filtering*, *enhancement*, *restoration*, dan *compression*. Tujuan dari *Image Processing* adalah untuk menyempurnakan citra dan mendapatkan informasi yang berguna (Maulana et al., 2023). Seperti memperjelas atau menghilangkan *noise* pada citra, atau mengubah

tampilan citra untuk memudahkan analisis. *Image Processing* sering digunakan dalam berbagai bidang, seperti pengolahan medis, pengenalan wajah, pengolahan citra satelit, pengenalan pola, dan lain-lain. Teknologi *Image Processing* sangat penting dalam pengembangan algoritma *Computer Vision*, termasuk dalam pengembangan algoritma *Deep Learning*.

Sebuah citra adalah representasi visual dari objek atau fenomena yang disajikan dalam bentuk gambar. Citra dapat disimpan dalam berbagai format *digital* dan dapat membawa banyak informasi (Adhinata et al., 2020). Sebuah citra dapat berisi informasi tentang warna, bentuk, ukuran, tekstur, dan banyak lagi. Informasi ini dapat diekstraksi dan dimanipulasi menggunakan teknik pengolahan citra seperti segmentasi citra, deteksi fitur, dan pengenalan pola. Citra adalah sumber daya yang sangat berharga dalam dunia komputasi visual dan penelitian terus dilakukan untuk meningkatkan teknik pengolahan citra yang ada.

## **2.6 *Computer Vision***

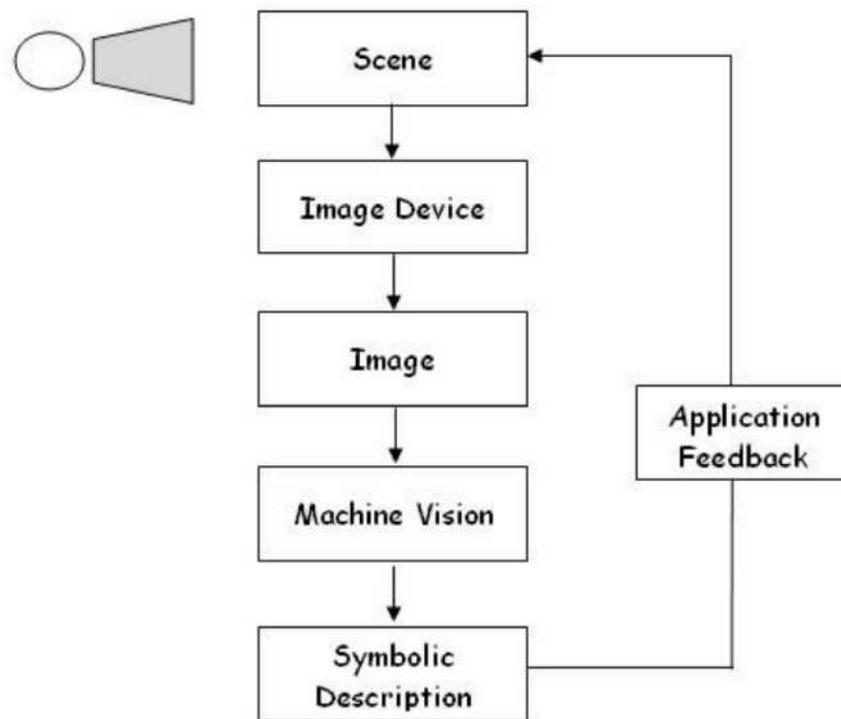
*Computer Vision* adalah bidang studi yang memungkinkan komputer untuk memproses, menganalisis, dan memahami gambar dan video dalam cara yang mirip dengan penglihatan manusia. Salah satu tujuan dari *Computer Vision* adalah untuk membuat komputer dapat "melihat" dunia seperti yang dilihat oleh manusia (Wahyudi & Kartowisastro, 2011). Konsep utama dalam *Computer Vision* adalah pengolahan citra, yang mencakup teknik untuk memanipulasi gambar untuk mendapatkan informasi yang berguna, seperti pengenalan objek, ekstraksi fitur, segmentasi objek, dan deteksi objek. Berikut adalah gambaran dari proses dalam visi komputer yang dapat dilihat pada gambar 2.1.



Gambar 2. 1 Proses pada Visi Komputer

*Computer Vision* dapat diklasifikasikan berdasarkan fungsinya menjadi dua kategori utama, yaitu inspeksi dan *sorting* (Neethu & Anoop, 2015). Inspeksi merupakan salah satu aplikasi *Computer Vision* yang digunakan untuk memeriksa kondisi fisik dan kemungkinan kecacatan pada sebuah objek. Contohnya adalah inspeksi pada produk manufaktur atau mesin, seperti memeriksa cacat pada produk atau memeriksa keandalan mesin pada suatu proses produksi. Sedangkan *sorting*, seperti namanya, digunakan untuk mengurutkan atau memisahkan objek-objek yang memiliki kriteria tertentu. Contohnya adalah memisahkan buah yang sudah matang atau tidak matang dalam industri pertanian, atau memisahkan sampah berdasarkan jenisnya di industri pengolahan sampah. Kedua kategori aplikasi ini memiliki beragam metode dan teknik yang berbeda untuk memproses data citra dan mencapai tujuan yang diinginkan.

Dalam bidang visi komputer, terdapat 7 elemen utama yang membentuk struktur dasar suatu sistem visi komputer, yaitu *light sources*, *scene*, *image device*, *machine vision*, *symbolic description*, dan *possible application feedback* (Amrizal & Aini, 2013). Berikut ini adalah struktur visual dari sistem visi komputer yang dapat dilihat pada gambar 2.2.



Gambar 2. 2 Struktur Visi Komputer

Adapun komponen dari *Computer Vision* sebagai berikut:

1. *Light source*, yang merupakan sumber cahaya yang digunakan dalam *Computer Vision*.
2. *Scene*, yaitu kumpulan dari objek-objek,
3. *Image device*, yaitu alat yang digunakan untuk mengubah letak benda yang direpresentasikan.
4. *Machine vision*, merupakan sistem yang menafsirkan gambar melalui ciri-ciri, pola maupun objek yang dapat ditelusuri oleh sistem.
5. *Symbolic description*, yaitu sistem yang dapat menggambarkan kinerja sistem kedalam simbol-simbol yang telah dimengerti oleh sistem.
6. *Possible application feedback*, merupakan suatu keadaan yang bisa memberikan respon menerima gambar dari suatu sistem penglihatan

## 2.7 Object Detection

*Object Detection* atau pengenalan objek adalah salah satu tugas mendasar dalam visi komputer (Vahab et al., 2019). *Object Detection* adalah salah satu cabang dari *Computer Vision* yang bertujuan untuk mendeteksi keberadaan dan

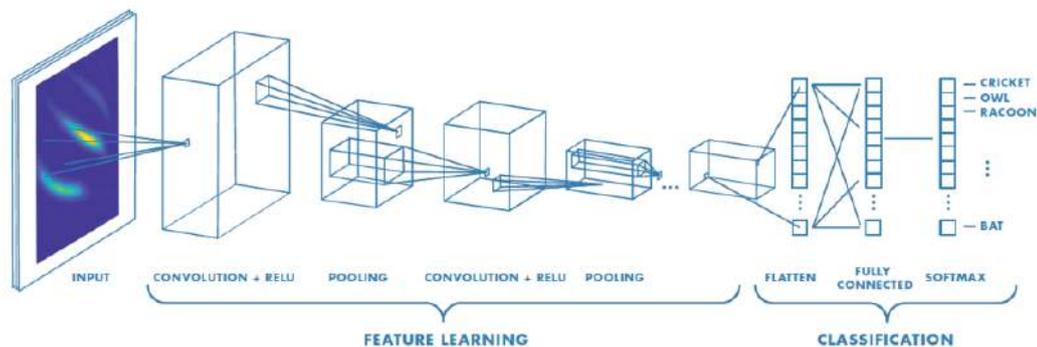
lokasi objek pada suatu gambar atau video. Teknik ini umumnya digunakan untuk melakukan pengenalan objek secara otomatis pada suatu gambar atau video. Tujuannya adalah untuk memberikan kemampuan pada mesin untuk mengenali objek seperti manusia, hewan, kendaraan, atau benda lainnya dalam gambar dan memperoleh informasi tentang kuantitas dan lokasi objek-objek tersebut. Pada dasarnya, *Object Detection* terdiri dari beberapa tahapan, yaitu:

1. *Preprocessing*: tahap awal yang dilakukan adalah mengubah gambar menjadi format yang dapat diproses oleh model *Deep Learning*, seperti mengubah ukuran gambar menjadi ukuran yang seragam atau normalisasi intensitas piksel.
2. *Feature extraction*: tahap ini melibatkan penggunaan algoritma *Deep Learning* untuk melakukan ekstraksi fitur pada gambar. Pada tahap ini, gambar akan diproses oleh beberapa lapisan jaringan saraf tiruan untuk mengekstrak fitur yang spesifik dan kompleks dari gambar.
3. *Region proposal*: tahap ini akan menentukan daerah-daerah pada gambar yang kemungkinan berisi objek, dengan cara memproses gambar menggunakan algoritma seperti *selective search* atau *region-based Convolutional Neural Network (R-CNN)*.
4. *Classification*: pada tahap ini, objek yang terdeteksi akan diklasifikasikan ke dalam kategori yang sudah ditentukan. Ini dilakukan dengan menggunakan teknik klasifikasi seperti *Convolutional Neural Network (CNN)* atau *Support Vector Machine (SVM)*.
5. *Post-processing*: tahap akhir dalam *Object Detection* adalah *post-processing*, di mana hasil deteksi objek yang sudah diklasifikasikan dianalisis untuk menghilangkan deteksi duplikat atau menggabungkan deteksi yang terpisah menjadi satu.

Dalam aplikasinya, *Object Detection* dapat digunakan dalam berbagai bidang, seperti otomotif, keamanan, transportasi, pengawasan, atau bahkan dalam industri kreatif seperti film dan videografi. Salah satu algoritma populer yang digunakan dalam *Object Detection* adalah YOLO (*You Only Look Once*).

## 2.8 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) atau dalam bahasa Indonesia sering disebut sebagai Jaringan Saraf Tiruan Konvulsional merupakan sebuah algoritma *Deep Learning* yang banyak digunakan untuk mengolah data gambar atau visual. CNN termasuk dalam kategori *Deep Neural Network* dikarenakan memiliki arsitektur dengan kedalaman yang tinggi (Putra, 2016). Umumnya, CNN banyak digunakan pada data citra untuk mempelajari dan mendeteksi fitur-fitur penting dalam gambar. Pada umumnya, *Neural Network* menggunakan data *array* satu dimensi sebagai *input*. Namun, pada algoritma CNN, data yang digunakan sebagai *input* berupa data dua dimensi (Ovtcharov et al., 2015). Berikut pada gambar 2.3 adalah gambaran dari arsitektur CNN.



Gambar 2. 3 Arsitektur CNN

CNN memiliki empat lapisan utama yang terdiri dari lapisan konvolusi (*Convolution Layer*), lapisan aktivasi (*Activation Layer*), lapisan penggabungan (*Pooling Layer*), dan lapisan terhubung penuh (*Fully-Connected Layer*) (Hibatullah, 2019). Berikut adalah penjelasan dari masing-masing *layer* yang ada pada CNN:

1. *Convolution layer*, adalah lapisan pertama dalam CNN yang melakukan konvolusi pada *input* gambar dengan sejumlah *filter* untuk menghasilkan *feature map*.
2. *Activation layer*, adalah lapisan yang mengaktifkan atau mematikan *neuron* pada *feature map* berdasarkan ambang batas tertentu.
3. *Pooling layer*, digunakan untuk mengurangi dimensi *feature map* dengan cara menggabungkan beberapa piksel menjadi satu nilai.

4. *Fully-connected layer*, adalah lapisan terakhir dalam CNN yang melakukan klasifikasi dengan menghubungkan *neuron* pada *layer* sebelumnya ke *neuron* pada *layer output*.

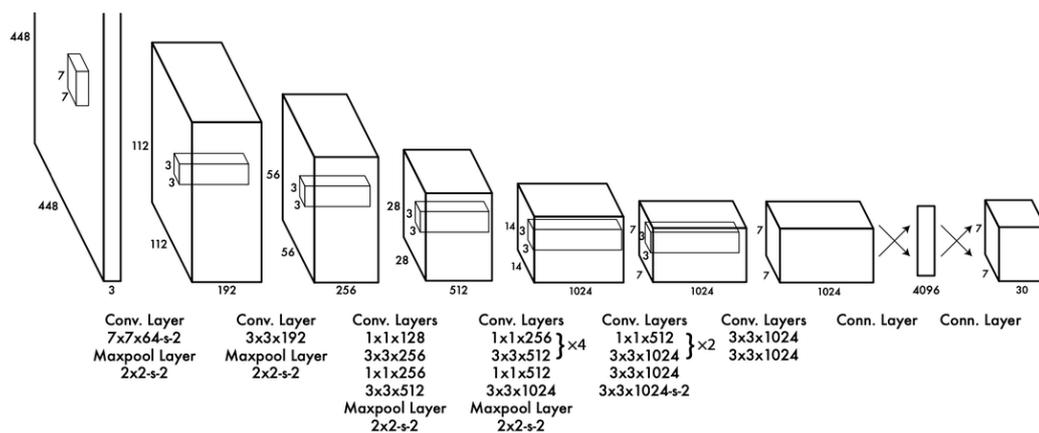
CNN bekerja dengan cara mempelajari *feature* pada gambar dengan cara melakukan konvolusi pada citra tersebut. Pada lapisan konvolusi, CNN menggunakan *filter* atau *kernel* yang berfungsi untuk memisalkan informasi pada citra secara bertahap, sehingga menghasilkan *feature map* yang merepresentasikan bagian-bagian penting pada citra. Selanjutnya, pada lapisan aktivasi, *feature map* yang dihasilkan pada lapisan konvolusi diproses dengan fungsi aktivasi untuk memperkuat sinyal yang relevan dan menghilangkan *noise* atau sinyal yang tidak relevan. Kemudian, pada lapisan *pooling*, *feature map* yang sudah diaktivasi akan dikompresi dengan melakukan operasi reduksi ukuran citra sehingga menjadikannya lebih efisien dan efektif. Setelah melalui beberapa lapisan konvolusi dan *pooling*, data yang sudah diolah kemudian diteruskan ke lapisan terhubung penuh atau *fully-connected layer*. Pada lapisan ini, CNN melakukan klasifikasi dan penentuan kelas berdasarkan *feature* yang sudah diekstrak sebelumnya.

Dengan demikian, secara keseluruhan cara kerja CNN adalah dengan mempelajari *feature* pada gambar secara bertahap dengan memanfaatkan lapisan konvolusi, aktivasi, dan *pooling* untuk menghasilkan representasi yang lebih efektif dan efisien. Selanjutnya, hasil representasi tersebut digunakan untuk melakukan klasifikasi pada lapisan *fully-connected layer*.

## 2.9 YOLO

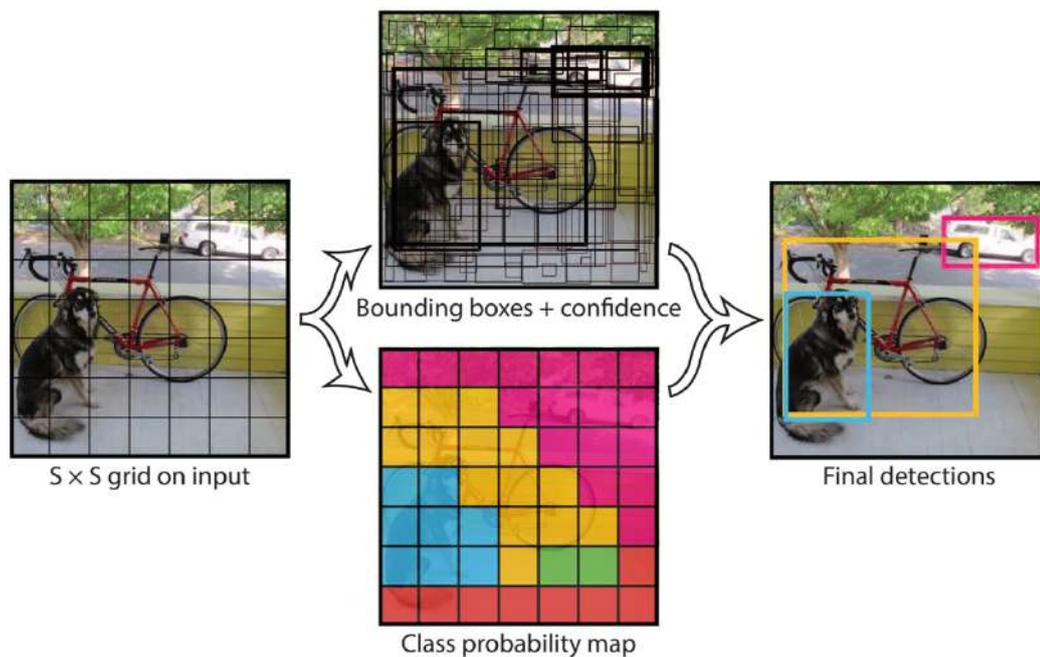
*You Only Look Once* (YOLO) adalah salah satu algoritma dalam *Computer Vision* yang digunakan untuk melakukan deteksi objek secara *Real-time* pada gambar maupun video. Inti dari algoritma YOLO adalah memiliki ukuran model yang kecil serta memiliki kecepatan perhitungan yang cepat (Jiang et al., 2022). Hal ini memungkinkan YOLO untuk melakukan deteksi objek secara *Real-time* dengan akurasi yang tinggi. Model YOLO menggunakan arsitektur *Convolutional Neural Network* (CNN) dengan banyak lapisan *convolutional*, yang memungkinkan model untuk mengekstraksi fitur gambar secara efisien.

Arsitektur YOLO asli terdiri dari 24 lapisan konvolusi yang diikuti oleh dua lapisan terhubung sepenuhnya (*fully connected layer*) (Jiang et al., 2022). Pada setiap lapisan konvolusi, YOLO menggunakan *filter* berukuran 3x3 dengan *stride* 1, dan dilakukan *zero-padding* pada tepi citra *input*. Setelah lapisan konvolusi, dilakukan lapisan *pooling* berukuran 2x2 dengan *stride* 2 untuk mengurangi dimensi citra *input* dan meningkatkan efisiensi komputasi. Pada bagian akhir, YOLO menghasilkan prediksi objek dengan menggunakan lapisan terhubung sepenuhnya (*Fully Connected Layer*) dengan *neuron* yang berkorespondensi dengan setiap *grid* pada citra *input*. Setiap *neuron* ini menentukan apakah suatu objek terdeteksi pada *grid* tersebut, dan memberikan informasi tentang ukuran dan lokasi objek relatif terhadap *grid*. Arsitektur dari YOLO dapat dilihat pada Gambar 2.4



Gambar 2. 4 Arsitektur YOLO (Redmon et al., 2016)

Proses deteksi objek pada YOLO dilakukan dengan membagi gambar ke dalam *grid-cell* yang sama besar, kemudian setiap *grid-cell* akan bertanggung jawab untuk mendeteksi objek yang ada di dalamnya. Selanjutnya, setiap *grid-cell* akan menghasilkan beberapa *bounding box* dengan ukuran dan posisi yang berbeda-beda, kemudian dilakukan *non-max suppression* untuk memilih *bounding box* yang paling sesuai dengan objek yang sebenarnya. Gambar 2.5 merupakan ilustrasi dari cara kerja YOLO



Gambar 2. 5 Ilustrasi Deteksi Objek YOLO (Redmon et al., 2016)

Salah satu keunggulan dari algoritma YOLO adalah kemampuan untuk melakukan deteksi objek secara *Real-time* dengan kecepatan yang tinggi. Dalam YOLO, deteksi objek dipandang sebagai masalah regresi, sehingga algoritma ini tidak memerlukan alur yang rumit. YOLO hanya menjalankan *Convolutional Neural Network* pada gambar untuk memprediksi deteksi objek. Oleh karena itu, YOLO dapat memproses citra secara *Real-time* dengan waktu latensi kurang dari 25 milidetik (Pramestya, 2018). Selain itu, YOLO juga memiliki kemampuan untuk mendeteksi objek dengan ukuran dan bentuk yang berbeda-beda, serta mampu mendeteksi objek yang saling tumpang tindih dengan akurat.

## 2.10 Vehicle Counting

*Vehicle Counting* atau penghitungan kendaraan merupakan sebuah teknik dalam *Computer Vision* yang dapat digunakan untuk menghitung jumlah kendaraan yang melintas pada suatu lokasi. Teknik ini biasanya dilakukan dengan menggunakan kamera CCTV yang dipasang di atas jalan raya atau lintasan kendaraan lainnya. Dalam *Vehicle Counting*, kamera akan merekam gambar kendaraan yang melewati lintasan tertentu, kemudian gambar tersebut akan diproses menggunakan teknik *Image Processing* dan *Computer Vision* untuk menghitung jumlah kendaraan yang telah melintas.

Teknik *Vehicle Counting* umumnya dilakukan dengan menggunakan algoritma *Deep Learning* seperti *Convolutional Neural Network* (CNN) dan *You Only Look Once* (YOLO). Dalam penggunaan algoritma YOLO, gambar dari kamera akan dibagi-bagi menjadi beberapa *grid*, kemudian setiap *grid* akan diproses menggunakan teknik CNN untuk mengenali kendaraan pada gambar tersebut. Hasil dari pengenalan tersebut kemudian akan digunakan untuk menghitung jumlah kendaraan yang melewati suatu lintasan pada suatu periode waktu tertentu.

*Vehicle Counting* sangat penting untuk banyak aplikasi dunia nyata, seperti manajemen lalu lintas (Zhang et al., 2017). Dengan menggunakan teknik ini, pihak berwenang dapat memantau dan menganalisis tingkat lalu lintas kendaraan pada suatu jalan, menentukan kecepatan rata-rata kendaraan, dan mengidentifikasi jam-jam sibuk yang memerlukan peningkatan kapasitas jalan atau tindakan pengaturan lalu lintas. Selain itu, teknik ini juga dapat digunakan untuk memantau kondisi lalu lintas pada waktu tertentu dan menentukan pola penggunaan jalan dalam jangka waktu tertentu.

### 2.11 *Confusion Matrix*

*Confusion Matrix* adalah alat penting dalam evaluasi klasifikasi yang digunakan untuk mengukur performa model dalam memprediksi kelas target. *Confusion Matrix* digunakan sebagai ukuran objektif untuk mengukur kualitas pendeteksian objek dengan akurat (Wu, 2022). *Confusion Matrix* berisi data tentang hasil klasifikasi yang sesungguhnya dan hasil klasifikasi yang diprediksi oleh model. Matriks ini terdiri dari empat komponen utama, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Pada Tabel 2.2 berikut merupakan contoh dari tabel *Confusion Matrix*.

Tabel 2. 2 Contoh Tabel *Confusion Matrix*

		Aktual	
		Kelas 1	Kelas 2
Prediksi	Kelas 1	TP	FP
	Kelas 2	FN	TN

Keterangan:

1. TP mewakili jumlah sampel yang benar-benar diklasifikasikan dengan benar sebagai positif.
2. TN mewakili jumlah sampel yang benar-benar diklasifikasikan dengan benar sebagai negatif.
3. FP mewakili jumlah sampel yang seharusnya diklasifikasikan sebagai negatif tetapi salah diklasifikasikan sebagai positif.
4. FN mewakili jumlah sampel yang seharusnya diklasifikasikan sebagai positif tetapi salah diklasifikasikan sebagai negatif.

*Confusion Matrix* memberikan gambaran yang jelas tentang sejauh mana model klasifikasi mampu membedakan kelas yang berbeda. Dengan menggunakan matriks kebingungan, kita dapat menghitung beberapa metrik evaluasi klasifikasi yang penting, seperti akurasi (*accuracy*), presisi (*precision*), *Recall* (*Recall*), dan nilai F1 (*F1-Score*). Berikut penjelasannya:

#### A. *Accuracy*

Akurasi (*Accuracy*) mengukur sejauh mana model mampu mengklasifikasikan dengan benar seluruh sampel. Dengan rumus sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad ..(2.1)$$

#### B. *Precision*

Presisi (*Precision*) mengukur sejauh mana model memberikan hasil positif yang benar dari seluruh hasil yang diprediksi positif. Dengan rumus sebagai berikut:

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{Total\ Prediksi\ Positif} \quad ..(2.2)$$

#### C. *Recall*

*Recall* mengukur sejauh mana model mampu mendeteksi secara benar semua sampel yang sebenarnya positif. Dengan rumus sebagai berikut:

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{Total\ Data\ Positif} \quad ..(2.3)$$

#### D. *F1-Score*

Nilai *F1-Score* merupakan ukuran gabungan dari *Precision* dan *Recall*.

Dengan rumus sebagai berikut:

$$F1 - Score = 2 \times \left( \frac{(Precision \times Recall)}{(Precision + Recall)} \right) \quad ..(2.4)$$