BAB II
DASAR TEORI

2.1 KAJIAN PUSTAKA

Berdasarkan riset yang telah dilakukan oleh Wawan Setiawan dan Naufal Hafidz Farhan pada tahun 2022 dengan judul "Deteksi Objek Plat Nomor Kendaraan Dengan Metode CNN". Pada riset ini membahas terkait bagaimana membangun sistem pendeteksi plat nomor mobil dengan memanfaatkan metode CNN, yang dilatar belakangi oleh tingginya kebutuhan masyarakat akan transportasi dapat memicu terjadinya kemacetan, yang disebabkan tata kelola parkir yang kurang tertata dan untuk membantu dalam proses mencatat nomor kendaraan dengan metode CNN. Pada penelitian ini menggunakan *dataset* berupa 240 citra plat nomor kendaraan bermotor lisensi Indonesia, dimana *dataset* diolah menjadi dua jenis yakni data *train* dan data *test*. *Input* yang digunakan pada riset ini yaitu berupa citra yang diambil melalui kamera *smartphone*. Langkah-langkah prosedur *investigasi* CRISPDM diikuti untuk menyelesaikan *investigasi* dan pemrosesan data. Dengan menggunakan perangkat lunak *Anaconda* dan bahasa pemrograman *Pytho*n, peneliti membuat model dari *dataset* plat nomor yang dikumpulkan. Data latih dan uji diambil dari *dataset* plat nomor dan digunakan untuk menganalisis angka dan huruf pada plat. Selain itu, data gambar plat nomor yang telah diperoleh dapat dipecah menjadi dibagi menjadi data *training* dan data uji *testing*[3].

Pada penelitian yang dilakukan oleh Syed Talha Abid Ali, Hakeem Usama, Ishtiaq Rasool Khan, Muhammad Murtaza Khan, dan Asif Siddiq pada tahun 2021 dengan judul "*Mobile Registration Number Plate Recognition Using Artificial Intelligence*". Dalam melakukan penelitian ini, memiliki tujuan yakni agar dapat menjabarkan bagaimana cara memasukan *Convolutional Neural Network* (CNN) ke dalam arsitektur atau rancangan sistem pendeteksi plat nomor kendaraan, dengan latar belakang penelitian ini karena adanya keterbatasan pada penelitian sebelumnya yang diakibatkan oleh latar belakang plat nomor, jarak konstan, dan sudut kamera. *Dataset* yang digunakan pada penelitian ini berisi 20 video, ditangkap oleh kamera yang dipasang dalam kondisi kendaraan bergerak. Kemudian dirancang algoritma pembingkaian yang mengekstrak bingkai dari video

yang diambil. Dengan demikian, 1000 *frame* diekstraksi dan disimpan dalam format *Portable Network Graphics* (PNG) dengan resolusi (1.920 × 1.080). Pada penelitian ini menggunakan *YOLO*v5 karena dianggap paling baik untuk deteksi objek. Model tersebut menggunakan CSP (*Cross Stage Partial*) *Networks* untuk mengekstrak fitur informatif dari setiap gambar *input* dan PANet untuk mendapatkan piramida fitur yang membuatnya lebih cepat untuk dideteksi dalam skenario *real time*. *Dataset* beranotasi dibagi menjadi 70% pelatihan, 20% validasi, dan 10% *dataset* pengujian dan dihasilkan akurasi deteksi 98%[4].

Sesuai hasil yang dipublikasikan pada tahun 2021 dengan judul "Rancang Bangun Sistem Deteksi Plat Nomor Kendaraan Berbasis *Computer Vision* Dengan Metode *Optical Character Recognition*" yang ditulis oleh Syamsul Falah Annur, Rais, dan Hepatika Zidny Ilmadina. Topik penelitian ini adalah sistem deteksi plat nomor kendaraan berbasis *Optical Character Recognition* yang dilatar belakangi oleh tingginya jumlah kecelakaan lalu lintas, yang diakibatkan karena menerobos lampu lalu lintas, menggunakan kecepatan yang cukup tinggi pada saat lampu merah, berhenti ditepi jalan, dan mengabaikan rambu-rambu lalu lintas. Berdasarkan *testing* yang telah dilakukan, diketahui *Optical Character Recognition* dapat mengenali plat nomor dari jarak aman di jalan raya. Dari hasil penelitian ini, OCR memiliki tingkat keberhasilan 53,04 % dalam mengartikan teks[5].

Pada penelitian yang dilakukan oleh Sohaib Abdullah, Md Mahedi Hasan, dan Sheikh Muhammad Saiful Islam pada tahun 2018 dengan judul "*YOLO-Based Three-Stage Network for* Bangla *License Plate Recognition in* Dhaka *Metropolitan City*". Penelitian ini membahas tentang bagaimana merancang sistem pengenalan plat nomor Bangla menggunakan *YOLO-Based Three-Stage Network* (*YOLOv3*) sebagai metode dengan latar belakang penelitian ini adalah banyaknya pelanggaran lalu lintas, kejahatan, serta kemacetan yang terjadi di kota Dhaka. Pada penelitian ini menggunakan *dataset* berupa 1.500 citra plat nomor kendaraan Bangladesh yang diperoleh manual dari jalan dengan berbagai kondisi dunia nyata dan dari sudut pandang yang berbeda. Sistem deteksi ini bersifat *real-time*. Metode yang diusulkan dievaluasi pada *dataset* ini dan dua tahap pertama dari arsitektur tiga tahap baru kami mencapai IoU lebih dari 85%untuk setiap kelas, dan model berbasis *ResNet*-20 kami tercapai 92.7% akurasi untuk pengenalan karakter. Arsitektur ALPR

efisien dalam hal mendeteksi dan mengenali sampel plat nomor yang sangat sulit dikenali[2].

2.2 DASAR TEORI

Penelitian ini memiliki tujuan untuk mengembangkan sistem pendeteksi plat nomor kendaraan dengan menggunakan sejumlah kerangka teori yang telah ditetapkan sebagai pedoman. Berikut beberapa teori yang dapat digunakan sebagai penunjang dalam proses perancangan pada kegiatan penelitian:

2.2.1 *Artificial Intelligence* (AI)

Kecerdasan buatan atau yang dikenal sebagai *Artificial Intelligence* (AI), adalah cabang ilmu komputer yang bertujuan untuk mereplikasi kemampuan pemecahan masalah otak manusia di dalam sebuah mesin. Fungsi umum AI adalah analisis dan klasifikasi data, seperti halnya otak manusia[6]. Beberapa ilmuwan yang kredibel, termasuk H. A. Simon (1987), yang dirujuk dalam buku "*Dasar Komputasi Cerdas Artificial Neural Network* (2016)" telah menawarkan pandangan mereka tentang apa yang dimaksud dengan Kecerdasan Buatan. Mereka setuju bahwa itu adalah studi, aplikasi, dan instruksi pemrograman komputer dengan tujuan menyelesaikan tugas yang tampak cerdas bagi manusia. Kelebihan *Artificial Intelligence* antara lain, sebagai berikut:

1. Bersifat permanen dan tidak berubah.
2. Dapat menyimpan berbagai informasi atau data tanpa adanya batasan
3. Cepat dan akurat dalam mengerjakan suatu pekerjaan.
4. Kemampuan memecahkan masalah yang kompleks.
5. Bisa menduplikasi dan mentransfer suatu kemampuan dengan mudah dari satu komputer ke komputer lain.
6. Penggunaan tanpa adanya batas waktu, karena *Artificial Intelligence* tidak mengenal lelah dan bosan.

Artificial Intelligence juga memiliki beberapa kekurangan antara lain, sebagai berikut:

1. Kecerdasan yang ada pada *Artificial Intelligence* tergantung pada apa yang di *input* oleh *programmer.*

2. Tidak memiliki *Common Sense. Common Sense* adalah kemampuan yang tidak hanya sekedar memproses sebuah informasi, melainkan mengerti akan informasi tersebut.
3. Tidak memiliki kemampuan mengembangkan pengetahuan, dan pengembangan pengetahuan pada *Artificial Intelligence* tergantung pada sistem yang dibangun.

Artificial intelligence bekerja sesuai dengan algoritma pemrograman pada sistem komputer yang diberikan dalam proses pembuatannya. Algoritma pemrograman digunakan sebagai kerangka berpikir dari *artificial intelligence* dalam memproses berbagai jenis data. Dilansir dari *Brookings*, algoritma pemrograman *artificial intelligence* memerlukan data yang banyak dan kuat agar komputer dapat membedakan pola yang baik. Dengan banyaknya data juga algoritma yang kompleks mesin seakan-akan dapat berpikir sendiri, membuat keputusan, belajar, juga beradaptasi.

2.2.2 *Deep Learning*

Deep learning (DL) merupakan algoritma atau bagian dari *artificial intelligence* (AI) dan *machine learning* (ML). *Deep learning* yaitu pengembangan dari *neural network multiple layer* yang digunakan dalam menyelesaikan permasalahan dan menjalankan suatu fungsi seperti deteksi objek, pengenalan suara, *translate language* serta fungsi lainnya yang mampu membuat keputusan akurat berdasarkan data. Untuk memecahkan masalah dan melakukan aktivitas seperti identifikasi objek, pengenalan suara, *translate language*, serta fungsi lainnya yang membutuhkan sistem untuk membuat penilaian yang benar berdasarkan data. Terdapat perbedaan antara *deep learning* dan *machine learning* yaitu pada *deep learning* secara otomatis melakukan pelabelan dari data seperti gambar, video atau *text* tanpa memberikan pelabelan secara manual atau bisa dikatakan secara otomatis seperti layaknya pemikiran manusia. *Deep learning* adalah jenis pembelajaran data yang menggunakan beberapa lapisan pemrosesan untuk membangun representasi data. LeCun, Bengio, dan Hinton (2015) berpendapat bahwa hasil pembelajaran algoritma yang tidak dilakukan secara *eksplisit* oleh manusia dan digunakan untuk membuat keputusan representasi data. Berbeda dengan metode *machine learning* mesin yang lebih konvensional,

pembelajaran mendalam dapat secara otomatis mewakili tipe data termasuk foto, video, dan teks tanpa memerlukan pemrograman khusus atau keahlian domain[7]. Digunakan beberapa *hyperparameter* pelatihan dalam parameter pengujian *Deep learning*. Parameter yang digunakan oleh pendekatan *deep learning* didasarkan pada penemuan Lu Yifei (2017). Parameter tersebut antara lain sebagai berikut[8]:

a. *Hidden layer*

   Dalam arsitektur *deep learning*, *hidden layers* digunakan untuk menyatakan total layer tersembunyi beserta *neuron* yang ada pada tiap layer untuk mendapatkan desain yang optimal dalam arsitektur *deep learning*.

b. *Epoch*

   Saat memproses pengumpulan data, *epoch* ini digunakan untuk menunjukkan jumlah iterasi yang harus dilakukan. *Epoch* adalah siklus yang digunakan *algoritme deep learning* untuk belajar dari kumpulan data pelatihan secara keseluruhan. Setiap siklus dilambangkan dengan nomor uniknya sendiri. Ketika algoritma *deep learning* telah menyelesaikan satu *epoch*, menandakan telah memperoleh pengetahuan dari *dataset* pelatihan secara keseluruhan (Satria Wibawa, 2017).

c. *Learning rate*

   Selama proses pelatihan, inilah salah satu parameter pelatihan yang akan digunakan untuk menentukan nilai koreksi bobot. Angka untuk tingkat pembelajaran ini berada di antara nol dan satu. Ketika ada *learning rate* yang lebih tinggi, proses pelatihan akan maju dengan kecepatan yang lebih cepat. Jika *learning rate* dinaikkan, akurasi jaringan akan turun, namun jika *learning rate* diturunkan, akurasi jaringan akan meningkat. Hal ini akan menjadikan proses pelatihan yang memakan waktu lebih lama.

d. *Evaluate Performance*

   Performa model diukur menggunakan metrik seperti *accuracy*, TPR/sensitivitas dan Kurva *Receiver Operating Characteristic* (ROC). Performa model tersebut ditentukan oleh *epoch, batch size*, dan *learning rate* yang dioptimalkan untuk metrik tersebut.

Perbedaan antara *machine learning* dan *deep learning* yaitu:

1. Pada sistem *machine learning*, manusia perlu mengidentifikasi dan memberi *hand code* untuk fitur yang diterapkan berdasarkan jenis data (misalnya, nilai piksel, bentuk, orientasi), sistem *deep learning* mencoba mempelajari fitur-fitur tersebut tanpa *intervensi* manusia tambahan.
2. Karena jumlah data yang sedang diproses dan kompleksitas perhitungan matematika yang terlibat dalam algoritma yang digunakan, sistem *deep learning* membutuhkan perangkat keras yang jauh lebih kuat daripada sistem *machine learning* yang lebih sederhana.
3. Karena ada begitu banyak parameter dan rumus matematika yang rumit yang terlibat, sistem *deep learning* dapat membutuhkan banyak waktu untuk berproses dibandingkan dengan *machine learning.*
4. Algoritma yang digunakan dalam *machine learning* cenderung mengurai data dalam beberapa bagian, kemudian bagian-bagian itu digabungkan untuk menghasilkan hasil atau solusi. Sedangkan *deep learning* melihat seluruh masalah atau skenario dalam satu proses. Misalnya, jika ingin membuat program untuk mengidentifikasi objek tertentu dalam gambar yaitu plat nomor pada mobil di tempat parkir, kita harus memprosesnya dalam dua langkah yaitu deteksi objek dan pengenalan objek pada *machine learning*. Sedangkan pada *deep learning*, dengan pelatihan program akan mengembalikan objek yang diidentifikasi dan lokasinya dalam gambar dalam satu hasil.

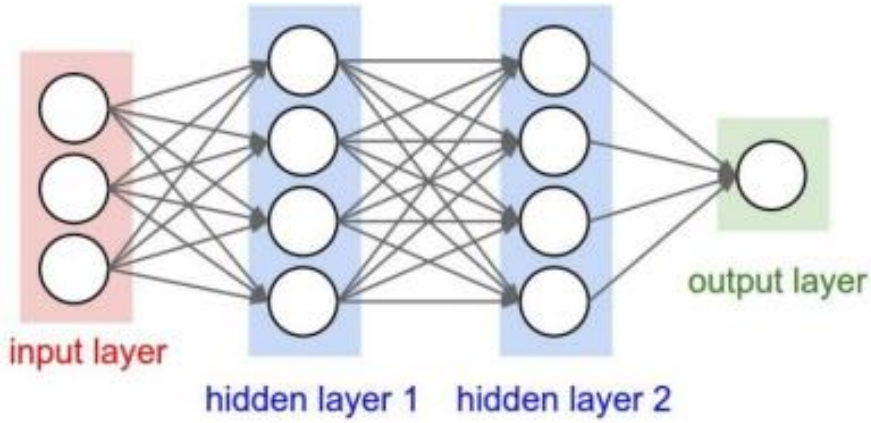
Jenis-jenis Algoritma pada *Deep Learning* yaitu:

- *Deep Neural Network* (DNN)
- *Artificial Neural Networks* (ANN)
- *Convolutional Neural Network* (CNN)

2.2.3 *Convolutional Neural Network* (CNN)

Convolutional Neural Networks (CNN) merupakan jaringan saraf buatan *feed-forward* yang dalam, dengan mempertahankan struktur hierarkis pada jaringan saraf, dalam melakukan pembelajaran untuk representasi fitur internal dan generalisasi fitur dalam masalah grafis umum seperti identifikasi objek dan masalah tampilan komputer lainnya[9]. *Convolutional Neural Network* (CNN) merupakan salah satu jenis *neural network* yang biasa digunakan pada data *image.* CNN bisa

digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image*[10]. *Convolutional Neural Network* dapat mencakup ratusan lapisan, dan masing-masing lapisan dilatih untuk mengenali aspek yang berbeda-beda dari sebuah gambar. Setiap gambar pelatihan akan diterapkan filter tersendiri pada resolusi yang berbeda, dan hasil gabungan kemudian dikirim ke lapisan pemrosesan berikutnya. Filter dapat dipengaruhi oleh kompleksitas dari kualitas yang sangat mendasar seperti kecerahan dan tepian, hingga fitur yang lebih canggih yang mencirikan item dengan cara tertentu. *Convolutional Neural Network* (CNN) terdiri dari *input* layer, *output* layer, dan *hidden layers* yang ada diantara *input* layer dan *output* layer[11]. Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.



Gambar 2. 1 Arsitektur CNN[10].

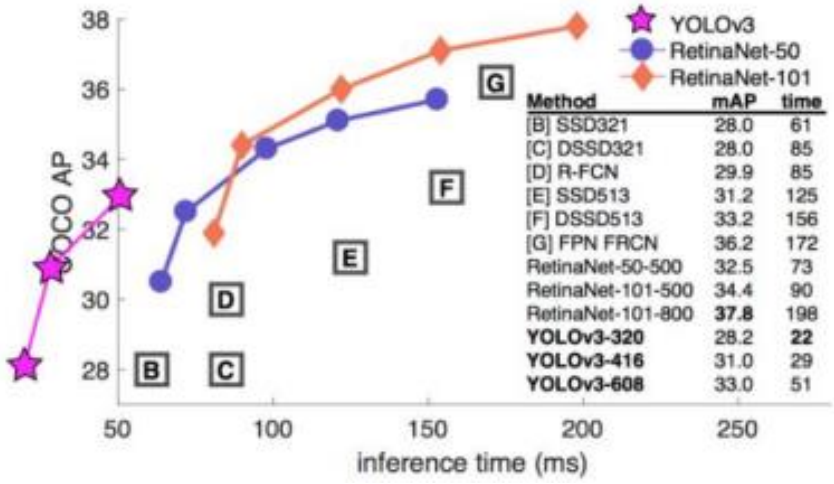
Sebuah MLP seperti pada Gambar 2.1. memiliki layer (kotak merah dan biru) dengan masing-masing layer berisi ji *neuron* (lingkaran putih). MLP menerima *input* data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan *output*. Setiap hubungan antar neuron pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Disetiap data input pada layer dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non linear yang disebut sebagai fungsi aktivasi.

CNN memiliki fungsi untuk melakukan ekstraksi fitur. Fitur-fitur perlu didapatkan guna proses atau tugas seperti klasifikasi, *clustering* ataupun regresi. Pada *machine learning* konvensional dilakukan ekstraksi fitur manual, artinya

ditentukan terlebih dahulu fitur-fitur yang diekstraksi. Sedangkan CNN melakukan ekstraksi fitur secara otomatis pada *convolutional* layer, *pooling* layer dan juga aktivasi *Rectified Linear* Unit (ReLU). Selanjutnya fitur-fitur dilakukan proses klasifikasi pada *Fully Connected* layer (FCL) dan aktivasi *softmax.* Arsitektur yang ada di *Convolutional Neural Network* yaitu antara lain, *AlexNet, VGG16, VGG19, GoogleLeNet, Inception-V3, ResNet50, ResNet101, InceptionResNetV2, Squeezenet, MobileNet*[12].

2.2.4 *YOLOv3*

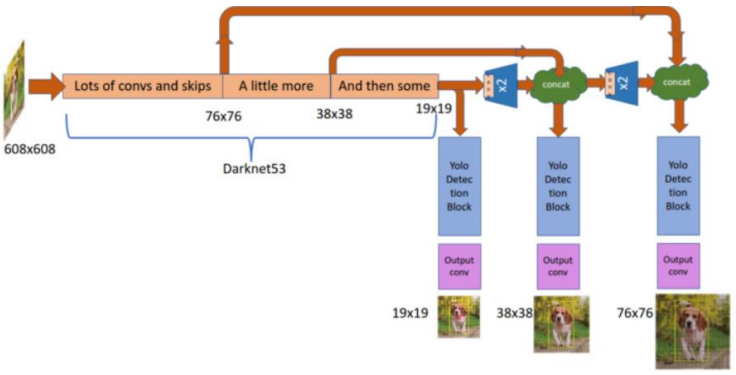
Sistem identifikasi objek yang disebut *You Only Look Once* (*YOLO*) berguna dalam proses bingkai pada deteksi objek dalam permasalahan hubungan tunggal dan memproses *pixel* citra kedalam kotak pembatas (*bounding box*) yang dipisahkan secara spasial serta probabilitas *class* terkait pemrosesan secara *realtime*[13]. *YOLO*v3 merupakan arsitektur *one-stage object detection*. Deteksi objek diperlakukan sebagai masalah regresi dalam desain identifikasi objek *one-stage*. Hal tersebut dapat memperkirakan *class probability* dan koordinat dari *bounding box* yang akan berisi objek dalam *single* step pada *input* gambar[14].



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

Gambar 2. 2 Performasi *YOLO*v3[15]**.**

Berdasarkan Gambar 2.2 jika dibandingkan dengan algoritma identifikasi objek lainnya, *YOLO*v3 menawarkan performa terbaik dan waktu pembelajaran tersingkat. Berdasarkan desain *Darknet* 53, jaringan *YOLO*v3 berisi 53 *convlutional layers*. Struktur *convolutional* dan *residual YOLO*v3 tersusun menjadi 53 *convlutional layers* yang disebut dengan *Darknet* 53[16].



Gambar 2. 3 Arsitektur *YOLOv3*[17].

Metode deteksi objek sebelum *YOLO* sudah sangat beragam antara lain seperti CNN, R-CNN, *Fast*-CNN. Pendeteksi objek terdahulu menerapkan model ke gambar di berbagai lokasi dan skala. Daerah penilaian yang tinggi pada gambar dianggap sebagai pendeteksian yang berhasil. *YOLO* berbeda dengan metode-metode sebelumnya. Menggunakan Neural Network, *YOLO* membagi gambar menjadi region kecil dan memprediksi *bounding boxes* dan probabilitas untuk masing-masing region[18]. Algoritma *YOLO* membagi gambar *input* yang diberikan ke dalam sistem grid SxS. Setiap penggilingan pada gambar *input* bertanggung jawab untuk deteksi pada objek. Sekarang sel kisi memprediksi jumlah kotak batas untuk suatu objek.

Pada proses pelatihannya, *YOLOv1* fokus pada penggunaan *multi-part loss function* (dirumuskan seperti rumus di atas) yang dapat merespon langsung terhadap performa dan keseluruhan model deteksi secara bersamaan. Selain itu, *YOLOv1* juga mengalami kesulitan dalam mengenali objek-objek kecil yang berdekatan yang diakibatkan oleh adanya spatial constraints pada prediksi bounding *box*-nya. *YOLOv1* juga mengalami kesulitan dalam mengenali objek-objek baru dari gambar yang memiliki konfigurasi maupun aspek rasio yang berbeda karena *YOLOv1* hanya mempelajari prediksinya dari data-data yang diberikan saja. Untuk memperbaiki batasan-batasan tersebut, Redmon, dkk., akhirnya merilis *YOLOv2* dan *YOLO9000* pada paper keduanya pada tahun 2016. Pada dasarnya *YOLOv2* memfokuskan pengembangan pada perbaikan *recall* dan lokalisasi bersamaan dengan mempertahankan akurasi klasifikasinya. *YOLOv2* menggunakan kustomisasi jaringan berbasis arsitektur *GoogleNet* yang memiliki kecepatan yang lebih cepat dari VGG-16, namun dengan akurasi yang lebih rendah. Selain itu,

penggunaan algoritma *open-source Darknet*19 juga mampu meningkatkan kinerja dan performa dari *YOLOv2*. *Darknet-19* memiliki prinsip kerja seperti *Network In Network* (NIN) yang memanfaatkan global *average pooling* untuk melakukan prediksi. Berdasarkan *YOLOv2* ini, Redmon, dkk., juga mengajukan model yang lebih kuat yang diberi nama *YOLO*9000. *YOLO*9000 menggunakan *Hierarchical View of Object Classification* untuk dapat menyatukan *dataset-dataset* yang berbeda. Selain itu, *YOLO*9000 juga menggunakan algoritma pelatihan gabungan yang memungkinkan untuk menggabungkan pelatihan data deteksi dan klasifikasi pada detektor. *YOLO*9000 menggunakan *dataset* label dari *WordNet* yang mampu mengenali 9000 kelas objek. Dikarenakan *WordNet* berbentuk grafik berarah yang susunannya cukup kompleks, maka untuk menyederhanakannya dibentuklah bentuk pohon hirarki yang diberi nama *WordTree* yang konsepnya berasal dari *ImageNet*. Kemudian *WordTree* diisi dengan dataset gabungan dari COCO dan *ImageNet. YOLO*9000 mampu mempelajari jenis objek baru namun masih kesulitan untuk dapat mempelajari kategori objek baru. Selanjutnya, pada tahun 2018, Redmon, dkk., berhasil meluncurkan versi terbaru dari *YOLO*[19]. Berdasarkan laporan teknologi yang dirilisnya, *YOLOv3* memiliki ukuran yang lebih besar dari versi-versi sebelumnya. Namun, *YOLOv3* ini masih memiliki performa yang lebih cepat serta akurasi yang lebih baik dibandingkan versi-versi sebelumnya. *YOLO* v3 menerapkan regresi logistik pada *bounding box*-nya untuk dapat mendeteksi keobjekan lebih baik. Selain itu penggunaan softmax digantikan oleh *Independent Logistic Classifier* karena *softmax* dinyatakan tidak berpengaruh langsung terhadap performa. Selain itu, *binary cross-entropy loss* juga digunakan saat training untuk memprediksi *class object.* Dibanding dengan *Darknet-19, Darknet-53* digunakan pada versi ini dikarenakan memiliki ukuran tertinggi pada floating point operation per detiknya yang berarti dapat memanfaatkan GPU lebih baik sehingga lebih efisien dan lebih cepat. Namun, dibalik performa yang lebih tinggi dari versi sebelumnya ini, *YOLOv3* masih mengalami beberapa kesulitan. Satu diantaranya yaitu kesulitan dalam mengenali objek-objek berukuran medium dan besar. Selain itu, *YOLOv3* juga sulit untuk dapat mensejajarkan *bounding box* dengan objek pada gambarnya. Oleh karena itu, walaupun *YOLOv3* memiliki kecepatan tertinggi

daripada versi-versi sebelumnya, namun *YOLOv3* lebih disarankan untuk dijalankan pada matriks deteksi lama dengan 0,5 IOU[20].

2.2.5 *Object Detection*

Object Detection adalah teknik untuk mengenali dan melokalisasi beberapa objek di dalam suatu citra. Dalam skenario ini, objek dipecah menjadi dua kategori, yaitu yang merupakan objek dan yang bukan objek. Jadi, konsep *Object Detection* secara sederhana yaitu dengan menganalisis suatu citra atau gambar dan mengklasifikasikan kedalam dua kategori tersebut, yaitu mana yang objek dan mana yang bukan objek (*background*)[21]. Sistem deteksi tersebut terbagi menjadi dua jenis yaitu *soft detection* dan *hard detection*. Dimana, saat menggunakan *soft detection* hanya dapat mendeteksi objek. Namun, ketika menggunakan *hard detection*, objek dan posisinya akan teridentifikasi[22].

2.2.6 *Tensorflow*

Tensorflow dibuat sebagai pustaka untuk mempelajari pembelajaran mesin dan jaringan saraf dalam oleh Tim *Google Brain* yang merupakan organisasi riset Mesin Cerdas Google. *Tensorflow* dapat memudahkan dalam menghitung berbagai ekspresi matematika di mana tantangannya terletak pada jumlah waktu yang diperlukan untuk melakukan perhitungan dengan menggunakan pendekatan dari bidang optimasi kompilasi aljabar komputasi. Fitur unggulan dari *tensorflow* adalah[23]:

1. Melakukan proses definisi, optimalisasi, serta cepat dalam proses efisien menghitung ekspresi matematis dengan menggunakan *array* multidimensi (*tensors*).
2. Memungkinkan pembuatan model *Machine Learning* dan *Artificial Neural Networks* (ANN) menggunakan pemrograman.
3. *Graphics Processing* Unit (GPU) bekerja secara transparan dalam mengotomatiskan memori dan pengoptimalan data yang sama.

Library dibangun dan dirancang untuk eksperimen cepat dengan jaringan saraf yang dalam dan kompatibel ketika dijalankan dengan CNTK, *TensorFlow*, dan *Theano*[24]. Cara kerja dari *tensorflow* yaitu dengan menerima *input* sebagai larik multidimensi yang disebut Tensor, *TensorFlow* memungkinkan untuk membuat grafik dan struktur aliran data untuk menentukan bagaimana data bergerak melalui

grafik. Ini memungkinkan untuk membuat diagram alur operasi yang dapat dilakukan pada *input* ini, yang berjalan dalam satu arah dan keluar dari yang lain.

2.2.7 *OpenCV*

Pengenalan wajah adalah salah satu bentuk perwujudan *Computer Vision* dalam mengenali sesuatu. *OpenCv* (*Open Computer Vision Library*) merupakan sebuah pustaka program yang dipergunakan secara bebas dan bersifat *open source*. Program ini ditujukkan untuk pengolahan citra dinamis secara *real time* dan memiliki lebih dari 2500 algoritma yang sudah teroptimasi. Pada *OpenCV* menerapkan metode *Computer Vision* yang memungkinkan komputer dapat melihat objek sama seperti manusia sehingga dapat mengambil keputusan dan melakukan aksi berdasarkan objek yang dideteksi. *OpenCV* dapat dijalankan pada *multi platform* sehingga dapat diterapkan secara luas pada gambar atau video seperti *face recognition, face detection, object tracking, road tracking*[25]. Kelebihan dari *library OpenCV* yaitu sebagai berikut:

a. Memiliki *library* dokumen yang cukup banyak
b. Dapat bekerja secara cepat pada computer yang menggunakan *processor intel*
c. Komputasi yang lebih ringan bila dibandingkan menggunakan Matlab dalam hal pengolahan citra digital (gambar)
d. Dapat bekerja secara *real time.*

OpenCV mempunyai banyak fitur yang dapat dimanfaatkan, berikut ini adalah fitur utama dari *OpenCV* antara lain[26] :

a. *Image* and Video I/O

   Dengan antar muka ini kita dapat membaca data gambar dari *file*, atau dari umpan video langsung. Dan juga dapat menciptakan *file* gambar maupun video.

b. *Computer Vision* secara umum dan pengolahan citra digital (untuk *low* dan *mid level* API)

   Dengan antar muka ini kita dapat melakukan *experimen* uji coba dengan berbagai standar algorima *computer vision*. Termasuk juga deteksi garis, tepi, pucuk, proyeksi *elips, image pyramid* untuk pemrosesan gambar multi skala, pencocokan *template,* dan berbagai *transform* (*Fourier, cosine diskrit, distance transform*) dan lain lain.

c. Modul *Computer Vision High* Level

Di dalam *OpenCV* juga termasuk kemampuan "*high level*", seperti kemampuan tambahan untuk deteksi wajah, pengenalan wajah, termasuk *optical flow.*

d. Metode untuk AI dan *Machine Learning*

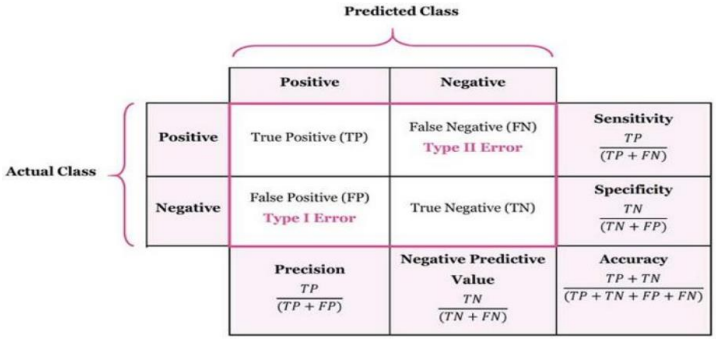
Aplikasi *computer vision* sering kali memerlukan *machine learning* atau metode AI lainnya, beberapa metode tersebut tersedia dalam paket *OpenCV machine learning.*

e. Sampling Gambar dan Transformasi

Di dalam *OpenCV* sudah terdapat antar muka untuk substraksi *subregion* dari gambar, *random sampling, rotating*, dan lain lain.

2.2.8. *Confusion Matrix*

Confusion Matrix merupakan suatu cara atau teknik yang dipakai dalam proses pengukuran kinerja atau tingkat kebenaran dari suatu metode klasifikasi. *Confusion matrix* merupakan tabel untuk melakukan perbandingan pada hasil klasifikasi berdasarkan hasil klasifikasi total data *testing* yang tepat dan data *testing* yang tidak tepat, dan usai diklasifikasikan dengan menggunakan sistem pada hasil klasifikasi secara aktual. Pada saat melakukan pengukuran kinerja dengan *confusion matrix,* digunakan empat kondisi untuk menggambarkan hasil proses klasifikasi yang dapat dilihat pada gambar 2.4.



Gambar 2. 4 *Confusion Matrix*[27].

Berdasarkan gambar 2.4, *confusion matrix* memiliki empat kondisi yaitu *True Positive* (TP), *False Negative* (FN), *False Positive* (FP), dan *True Negative* (TN). *True Positive* (TP) merupakan kondisi di mana prediksi yang diperoleh suatu model adalah benar dan sesuai dengan kondisi aktual. *False Negative* (FN) merupakan kondisi dimana prediksi yang diperoleh suatu model adalah salah atau tidak terjadi, namun kondisi aktualnya benar atau terjadi. Kodisi tersebut dianggap sebagai kesalahan atau Tipe II *Error*. *False Positive* (FP) merupakan kondisi

dimana prediksi yang diperoleh suatu model adalah positif atau benar, namun dalam kondisi aktual bernilai negatif atau salah. Kondisi ini juga dianggap sebagai kesalahan, namun hanya tergolong Tipe I *Error*. Dapat diartikan bahwa antara *False Negative* dan *False Positive*, lebih baik *False Positive*. *True Negative* (TN) merupakan kondisi dimana prediksi yang diperoleh suatu model adalah *negative* dan sesuai dengan kondisi *actual*, atau dapat diartikan model memprediksi dengan benar. Dengan adanya kondisi-kondisi tersebut dapat dilakukan evaluasi dengan menghitung nilai Sensitivitas, Spesifisitas, Akurasi, Nilai Prediktif Negatif, dan Presisi. Dimana sensitivitas yang biasa disebut *True Positive Rate* atau *Recall* merupakan rasio perbandingan prediksi benar positif dengan keseluruhan kondisi yang bernilai positif atau benar. Spesifisitas juga dikenal sebagai *True* Negative *Rate* merupakan kebenaran memprediksi negatif dibandingkan dengan keseluruhan data negatif. Akurasi merupakan rasio prediksi benar yaitu kondisi *True Positive* dan *True Negative* dengan keseluruhan data. Nilai Prediktif Negatif merupakan hasil yang diberi label dengan benar sebagai salah. Presisi merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf. *F1-Score* merupakan perbandingan rata-rata dari hasil presisi atau rasio prediksi benar positif dan *recall* yang dibobotkan [27].