

## **BAB II LANDASAN TEORI**

### **2.1 Tinjauan Pustaka**

Pada penelitian ini akan digunakan tinjauan pustaka sebagai referensi atau acuan dasar untuk membantu dalam menyelesaikan penelitian, berikut ini adalah tinjauan pustaka yang diambil:

#### 2.1.1 Tinjauan Terhadap Literatur 1

Oleh (Triano, Bansode *et al.*, 2018) melakukan penelitian dengan judul “Implementasi *Deep Learning* Untuk *Image Classification* Menggunakan *Convolutional Neural Network* (CNN) Pada Citra Wayang Golek” dalam penelitian tersebut melakukan klasifikasi tiga kelas gambar wayang golek, yaitu Cepot, Gatotkaca dan Semar dengan menggunakan algoritma *Convolutional Neural Network* (CNN), proses utama dalam pembuatan model ini diawali dengan proses training data. Proses training menggunakan *packages Keras* pada *python* dengan *back-end Tensorflow*. Menggunakan Skenario perbandingan jumlah data training dan testing (70 : 30, 80 : 20, dan 90 : 10) didapatkan tingkat akurasi terbaik menggunakan 90 :10 dengan jumlah data training dan testing 270:30 dengan akurasi 100 %.

#### 2.1.2 Tinjauan Terhadap Literatur 2

Oleh (Irfan dan Widodo, 2020) melakukan penelitian dengan judul “Penerapan Metode *Deep Learning Convolution Neural Network* pada Robot KRSBI Humanoid R-SCUAD” penelitiannya membahas tentang *Convolutional Neural Network* atau CNN yang dirancang memiliki 6 *hidden layer* dengan implementasi pada program robot menggunakan *library Tensorflow*. Gambar yang

diambil digunakan dalam proses training memiliki 9 jenis gambar berdasarkan tempat pengambilan gambar. Tiap jenis gambar terbagi menjadi 2 class yaitu 2000 gambar untuk class objek bola dan 2000 gambar untuk class objek bukan bola. Pengujian dilakukan secara *real time* dengan menggunakan bola berwarna putih di atas rumput hijau. Dari perancangan arsitektur dan hasil pengujian pendeteksian bola putih didapatkan persentase keberhasilan 67 % yaitu 5 dari sembilan model berhasil mengenali bola. Model dapat mengenali objek dengan kecepatan pengolahan gambar adalah maksimal 13 FPS.

### 2.1.3 Tinjauan Terhadap Literatur 3

Oleh (Maulana dan Rochmawati, 2020) melakukan penelitian dengan judul “Klasifikasi Citra Buah Menggunakan *Convolutional Neural Network*” membahas tentang *Deep Learning* menggunakan algoritma yang dibuat berdasarkan pada hukum matematik yang bekerja layaknya otak manusia. Salah satu pemanfaatan dari *deep learning* adalah dalam bidang *image processing* atau pengolahan citra digital. Algoritma CNN terdiri dari 3 layer utama yaitu *Convolutional Layer*, *Pooling Layer*, dan *Fully Connected Layer*. Pada penelitian ini menggunakan arsitektur CNN dengan perpaduan 3 *Convolutional Neural Network* dan 2 *Fully Connected Layer*. Pada tahap pembuatan system klasifikasi yang menggunakan *deep learning* terdapat beberapa tahapan proses utama yaitu pengumpulan data, perancangan system, *training*, dan *testing*. Dataset yang diolah adalah dataset citra buah-buahan yang berasal dari dataset *Fruit-360*. Kelas data yang digunakan yaitu sejumlah 15 kelas dari 111 kelas pada dataset *fruit-360*. Hasil dari proses *learning* didapatkan model CNN dengan akurasi 100% dan loss sebesar 0,012. Pada proses pengujian model CNN yang menggunakan 45 sampel citra buah didapatkan akurasi

sebesar 91,42%. Sehingga dapat disimpulkan bahwa metode CNN yang dirancang pada penelitian ini dapat mengklasifikasi citra dengan baik.

#### 2.1.4 Tinjauan Terhadap Literatur 4

Oleh (Javadi *et al.*, 2018) melakukan penelitian dengan judul “*Humanoid Robot Detection using Deep Learning: A Speed-Accuracy Tradeoff*” kemajuan terbaru dalam visi komputer telah membuat deteksi landmark di lapangan sepak bola lebih mudah untuk team. Namun, deteksition robot lain juga merupakan kemampuan penting yang belum dikumpulkan banyak perhatian di komunitas RoboCup sejauh ini. Masalah ini baik-baik saja diwakili dalam *RoboCup Soccer* dan *Rescue Robot Leagues* yang berbeda. Dalam makalah ini, kami membandingkan beberapa sistem deteksi dua tahap berbasis di berbagai *Convolutional Neural Networks* (CNN) dan sorot *trade off* kecepatan-akurasi. Pendekatan ini melakukan segmentasi citra berbasis tepi. Segmentation untuk mengurangi ruang pencarian dan kemudian CNN memvalidasi deteksi pada tahap kedua. Kami menggunakan gambar humanoid yang berbeda robot untuk melatih dan menguji tiga arsitektur CNN yang berbeda. Sebagian dari ini gambar dikumpulkan oleh team kami dan akan tersedia untuk umum. Eksperimen ini menunjukkan kemampuan beradaptasi yang kuat dari CNN yang lebih dalam model, dilatih pada robot yang terbatas, dapat berhasil membedakan jenis robot humanoid yang tidak terlihat dari wilayah non-robot (Javadi *et al.*, 2018).

#### 2.1.5 Tinjauan Terhadap Literatur 5

Oleh (Gustian Cahyadi, tanpa tanggal) melakukan penelitian dengan judul “Klasifikasi blok Lapangan untuk mengetahui posisi robot pada team Krakatau FC menggunakan *Convolutional Neural Network*” penelitian ini mengajukan judul

Klasifikasi Blok Lapangan untuk mengetahui posisi Robot pada Tim Krakatau FC menggunakan *Convolutional Neural Network* supaya dapat menentukan posisi robot berada secara akurat, data yang diperoleh dari objek lapangan berwarna hijau secara penuh, dari objek tersebut yang akan menentukan posisi robot. Hasil dari penelitian tersebut akan diterapkan ke Robot Krakatau fc. Dengan tujuan untuk menerapkan metode learning agar bisa mendeteksi klasifikasi blok lapangan dan untuk mengetahui seberapa akurat metode Tensorflow Lite untuk diimplementasikan pada robot Krakatau FC. Melakukan testing 300 data menggunakan model Edge TPU dan CPU, dengan mempertimbangkan *precision*, *recall*, dan Akurasi dan menyimpulkan bahwa Edge TPU Lebih unggul dan cepat dalam proses learning dibandingkan CPU.

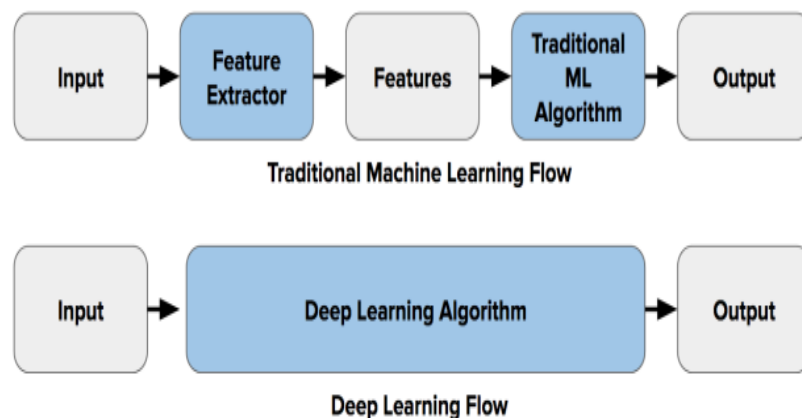
## 2.2 *Artificial Intelligence*

*Artificial Intelligence* atau Kecerdasan buatan merupakan suatu cabang dalam bidang sains komputer Sains yang mengkaji tentang bagaimana untuk melengkapi sebuah komputer dengan kemampuan atau kepintaran seperti manusia. Sebagai contoh, bagaimana komputer bisa belajar sendiri dari pengalaman dan data-data yang telah dikumpulkannya, bagaimana komputer mampu berkomunikasi dan mengucapkan kata demi kata. Dengan kemampuan ini, diharapkan komputer mampu mengambil keputusan sendiri untuk berbagai kasus yang ditemuinya. Industri AI ini berkembang semenjak tahun 1980an, meskipun sudah dimulai dari tahun 1970an Evolusi AI berjalan dalam dua jalur yang berbeda. Pertama, untuk menciptakan sistem komputer yang meniru proses berpikir manusia untuk menyelesaikan permasalahan umum. Misalnya program permainan catur. Kedua, mengkombinasikan pemikiran terbaik para ahli pada sepotong *software* yang

dirancang untuk memecahkan persoalan yang spesifik. Biasanya disebut juga dengan *expert system*, atau sistem pakar. Misalnya bagaimana seorang dokter menentukan penyakit seseorang, mulai dari tanya jawab, pemeriksaan kondisi tubuh seperti mata, tekanan darah, suhu tubuh dan sebagainya. Langkah-langkah ini pula yang berusaha diterapkan ke komputer yang mampu berpikir seperti pakar tersebut (Sutojo, T; Mulyanto, Edi; Suhartono, 2011).

### 2.3 *Deep Learning*

*Deep learning* (DL) adalah teknik dalam *Neural Network* (NN) yang menggunakan teknik tertentu seperti mesin *Boltzman* terbatas (RBM) untuk mempercepat proses pembelajaran di NN yang menggunakan beberapa lapisan atau lebih dari 7 lapisan. Dengan DL, waktu yang dibutuhkan untuk pelatihan dipersingkat, karena masalah hilangnya gradien dalam propagasi mundur lebih sedikit. Berbagai jenis DL antara lain *deep automatic coders*, *deep belief network*, *convolutional NN*, dan lain-lain (Ahmad, tanpa tanggal).



Gambar 2. 1 *Deep Learning Flow*

## 2.4 *Computer Vision*

*Computer Vision* adalah kemampuan komputer untuk melihat suatu objek, menampilkan objek digital, dan mengumpulkan data secara visual. Komputer dapat melakukan beberapa pekerjaan yang tidak dapat dilakukan manusia:

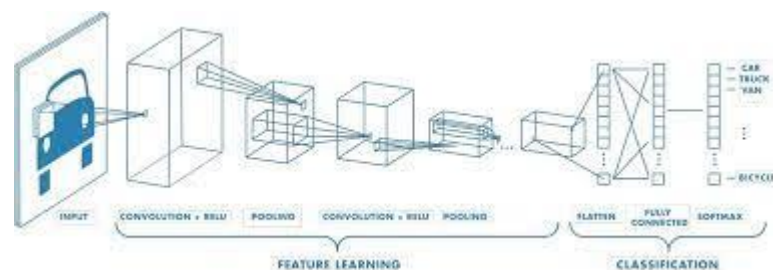
1. *Computer* mampu melihat data dalam bentuk pixel bahkan dalam warna yang berbeda.
2. *Computer* mampu membandingkan dua objek gambar yang persis.
3. *Computer* mampu melihat objek data selama berjam-jam bahkan lebih, *vision* itu sendiri adalah suatu proses evaluasi sebuah data yang bersumber dari image, dengan teknik *ekstrasi* menggunakan algoritma tertentu (Purno dan Wibowo, 2016).

## 2.5 **Klasifikasi Citra**

Klasifikasi citra adalah memasukkan gambar dan menetapkannya ke dalam kategori. Ini adalah salah satu masalah *Computer Vision*, yang dapat disederhanakan dan memiliki beberapa kegunaan. Salah satu kegunaan dari klasifikasi citra adalah klasifikasi tempat pada sebuah citra. Dengan klasifikasi, setiap gambar yang dimasukkan ke dalam set pelatihan diberi label. Selama klasifikasi, label dibandingkan dengan hasil hipotesis model pembelajaran dan memberikan nilai kesalahan. Klasifikasi terawasi bisa sangat efisien dan akurat dalam mengklasifikasikan citra tempat dan objek lain. Banyak metode dan algoritma yang dapat mendukung proses klasifikasi terawasi, terutama dengan teknik deep learning (Danukusumo, 2017).

## 2.6 Convolutional Neural Network (CNN)

*Convolutional neural network* merupakan salah satu jenis neural network yang umum digunakan dalam pengolahan data citra. Konvolusi atau yang biasa disebut dengan konvolusi adalah matriks yang berfungsi untuk menyaring citra. Jaringan saraf *convolutional* memiliki beberapa lapisan yang digunakan untuk menyaring setiap proses. Proses tersebut dikenal dengan proses pembentukan. Ada 3 tahapan dalam proses pelatihan, yaitu *convolution layer*, *pooling layer*, dan *fully connected layer* (Santoso dan Ariyanto, 2018).



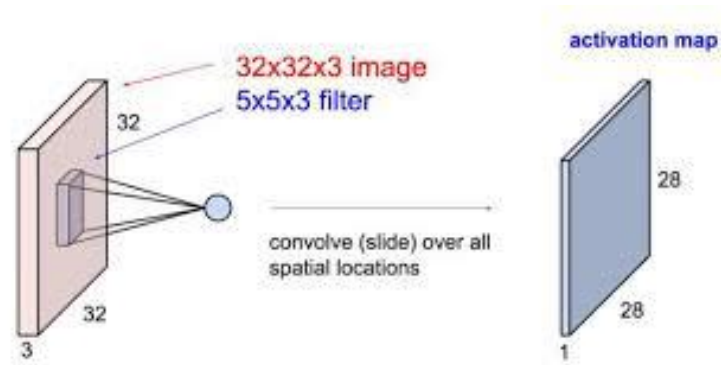
Gambar 2. 2 *Convolutional Neural Network*

(Sumber: <https://www.mathworks.com/discovery/convolutional-neural-network.html>)

### 2.6.1 Convolutional Layer

*Convolutional* mengalami proses konvolusi. Lapisan mengubah setiap filter menjadi bagian lengkap dari data masukan dan menghasilkan peta aktivasi atau peta fitur 2D. Filter yang terdapat pada *convolutional layer* memiliki panjang, tinggi dan ketebalan yang sesuai dengan input data channel. Setiap filter tunduk pada operasi shift dan "titik" antara data input dan nilai filter. Lapisan konvolusi mengalami kompleksitas model secara signifikan dengan mengoptimalkan outputnya. Ini

dioptimalkan untuk tiga parameter, kedalaman, panjang langkah, dan pengaturan zero padding (O'Shea dan Nash, 2015).



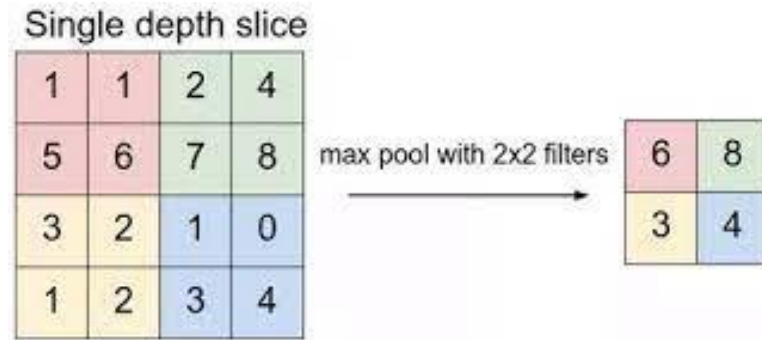
Gambar 2. 3 *Convolutional Layer*

(Sumber <https://www.google.com/journal-computing.org>)

### 2.6.2 *Pooling Layer*

*Pooling Layer* merupakan tahap setelah *Convolutional Layer*. *Pooling Layer* terdiri dari sebuah filter dengan ukuran dan stride tertentu. Setiap pergeseran akan ditentukan oleh jumlah *stride* yang akan digeser pada seluruh *area feature map* atau *activation map*. Dalam penerapannya, *pooling Layer* yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Sebagai contoh, apabila kita menggunakan *Max Pooling*  $2 \times 2$  dengan Stride 2, maka pada setiap pergeseran filter, nilai yang diambil adalah nilai yang terbesar pada area  $2 \times 2$  tersebut, Sedangkan *Average Pooling* akan mengambil nilai rata-rata.

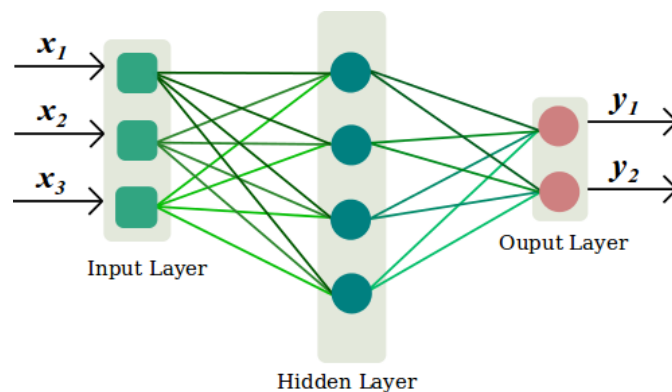


Gambar 2. 4 *Pooling Layer*

([https://www.researchgate.net/figure/A-pooling-layer-with-2-by-2-filter-size\\_fig5\\_343987422](https://www.researchgate.net/figure/A-pooling-layer-with-2-by-2-filter-size_fig5_343987422))

### 2.6.3 *Fully Connected Layer*

*Feature map* yang dihasilkan pada tahap sebelumnya berupa *multidimensional array*. Sehingga, sebelum peta fitur memasuki *fase Fully Connected Layer*, dia melewati proses “*Flatten*” atau pembentukan kembali (*Reshape*). Proses *Flatten* menghasilkan vektor yang digunakan sebagai input dari *Fully Connected Layer*. *Fully Connected Layer* memiliki beberapa *Hidden Layer*, *Action Function*, *Output Layer* dan *Loss Function*.

Gambar 2. 5 *Fully Connected layer*

([https://www.gabormelli.com/RKB/Fully-Connected\\_Neural\\_Network](https://www.gabormelli.com/RKB/Fully-Connected_Neural_Network))

#### 2.6.4 *Dropout*

*Dropout* merupakan upaya untuk mencegah *overfitting* dan mempercepat proses pembelajaran. *Overfitting* merupakan kondisi dimana hampir semua data yang telah melalui proses pelatihan mencapai persentase yang baik, namun terdapat ketidaksesuaian dalam proses prediksi. Atrisi menghilangkan sementara neuron sebagai lapisan tersembunyi atau lapisan yang terlihat dalam jaringan (Santoso dan Ariyanto, 2018).

### 2.7 *TensorFlow*

*TensorFlow* adalah pustaka perangkat lunak yang dikembangkan oleh team *Google Brain* dalam organisasi penelitian *Google Intelligent Machine* untuk melakukan pembelajaran mesin dan penelitian jaringan saraf dalam. Kemudian, *TensorFlow* menggabungkan teknik pengoptimalan kompilasi aljabar komputasional yang menyederhanakan penghitungan banyak ekspresi matematika di mana masalahnya adalah waktu yang dibutuhkan untuk melakukan penghitungan (Husada, 2019). Fitur utamanya meliputi:

1. Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan array multi dimensi (tensors).
2. Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin.
3. Penggunaan GPU yang transparan, mengotomatisasi manajemen dan optimalisasi memori yang sama dan data yang digunakan. *Tensorflow* bisa menulis kode yang sama dan menjalankannya baik di CPU atau GPU. Lebih

khusus lagi, *Tensorflow* akan mengetahui bagian perhitungan mana yang harus dipindahkan ke GPU.

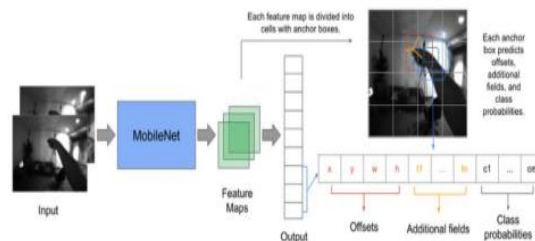
4. Skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar.

## 2.8 *Mobilenet v2*

Arsitektur *Mobilenet v2* merupakan penyempurna dari *Mobilnet*, umumnya *Mobilenet* dan CNN terdapat perbedaan pada penggunaan lapisan layer. *Convolution* layer pada *Mobilenet v2* menggunakan penyesuaian ketebalan filter dari input image. *Mobilenet* adalah model kecil latensi rendah daya rendah yang diukur untuk memenuhi batasan sumber daya dari berbagai kasus penggunaan. Menurut makalah penelitian, *Mobilenet v2* meningkatkan kinerja model seluler yang canggih pada banyak tugas dan tolak ukur serta pada seluruh spektrum ukuran model yang berbeda. *MobileNet V2* merupakan ekstraktor fitur yang sangat efektif untuk deteksi dan segmentasi objek Misalnya untuk deteksi, saat dipasangkan dengan *Single Shot Detector Lite*, *Mobilenet v2* sekitar 35 persen lebih cepat dengan akurasi yang sama dari *Mobilenet v1*.

*Bottleneck* dari *Mobilenet v2* mengkodekan input dan output perantara sementara lapisan dalam merangkum kemampuan model untuk mengubah dari konsep tingkat yang lebih rendah seperti piksel ke deskriptor tingkat yang lebih tinggi seperti kategori gambar. Dengan koneksi residual tradisional, pintasan memungkinkan pelatihan yang lebih cepat dan akurasi yang lebih baik. *Mobilenet v2* masih menggunakan *depthwise* dan *pointwise convolution*. *Mobilenet v2* menambahkan dua fitur baru yaitu *linear bottleneck*, dan *shortcut connections* antar *bottlenecks*. Pada bagian *bottleneck* terdapat input dan output antara model

sedangkan lapisan atau layer bagian dalam meng-enkapsulasi kemampuan model untuk mengubah input dari konsep tingkat yang lebih rendah (piksel) ke deskriptor tingkat lebih tinggi (Nufus *et al.*, 2021).



Gambar 2. 6 Mobilenet v2 (Sandler et al., 2018)

## 2.9 Loss

*Loss* adalah fungsi yang menggambarkan kerugian yang berkaitan dengan *probabilitas* yang dihasilkan oleh model CNN. *Loss function* yang digunakan dalam penelitian ini adalah *categorical cross-entropy*. Salah satu syarat *categorical crossentropy* dapat digunakan adalah pada proses klasifikasi menggunakan fungsi aktivasi *softmax*. *Categorical cross-entropy* digunakan saat hanya terdapat satu hasil yang benar. *Categorical cross-entropy* akan membandingkan distribusi dari probabilitas prediksi dengan distribusi dari kelas yang benar (target), dimana probabilitas dari kelas yang benar diset dengan 1 dan untuk kelas yang lain 0 (Maulana dan Rochmawati, 2020).

## 2.10 Convusion Matrix

Merupakan salah satu metode yang digunakan untuk mengevaluasi kinerja dari algoritma klasifikasi. Metode *confusion matrix* ini memiliki beberapa parameter penilaian terhadap kinerja dari model klasifikasi yang dihasilkan adalah:

1. *Precision* Merupakan parameter penilaian yang menghitung nilai rata-rata *precision* dari data hasil klasifikasi yaitu jumlah data yang benar antara nilai sebenarnya dengan hasil prediksi model yang dapat dihitung dengan rumus seperti pada persamaan berikut:

$$Precision = (TP) / (TP+ FP)$$

2. *Recall* Parameter penilaian yang didapat dari jumlah data benar seberapa banyak data yang keluar dalam hasil klasifikasi melalui rumus pada persamaan berikut:

$$Recall = (TP) / (TP+FN)$$

3. *Accuracy* Parameter penilaian untuk menguji akurasi model dan tingkat confident dalam melakukan klasifikasi terhadap data baru. Untuk mengetahui akurasi suatu model dapat dicari dengan rumus seperti pada persamaan berikut:

$$Akurasi = (TP+TN) / (TP+TN+FP+FN)$$