

# Algoritma Convolutional Neural Network Secara Komprehensif

**Ihsanul Fikri Abiyyu, Muhammad Azhar Rasyad, Muhammad Rafi Shiddiq,  
Muhammad Rizki Herfian**

Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF)

Jl. Lenteng Agung Raya No.20, Jakarta Selatan 12640

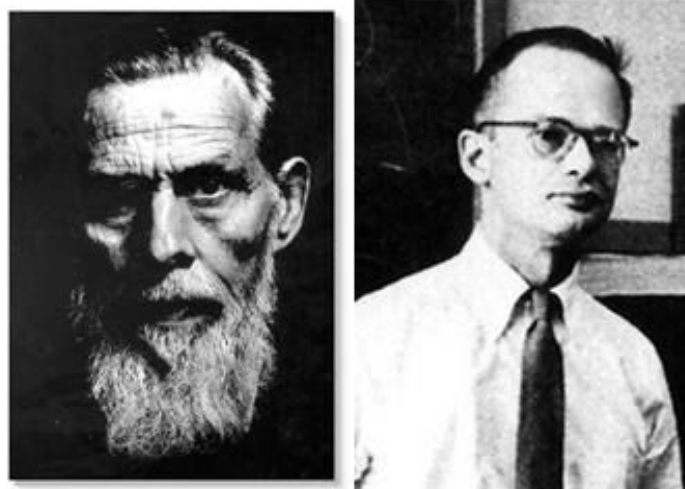
E-mail: [ihasnulcrz@gmail.com](mailto:ihasnulcrz@gmail.com), [muhazharrasyad@gmail.com](mailto:muhazharrasyad@gmail.com), [mrafishiddiq99@gmail.com](mailto:mrafishiddiq99@gmail.com),  
[iyanvermilian@gmail.com](mailto:iyanvermilian@gmail.com)

---

## 1. History

### a. Neural Network

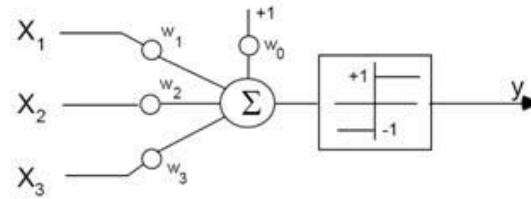
Perkembangan ilmu Neural Network sudah ada sejak tahun 1943 ketika Warren McCulloch dan Walter Pitts memperkenalkan perhitungan model neural network yang pertama kalinya. Mereka melakukan kombinasi beberapa processing unit sederhana bersama-sama yang mampu memberikan peningkatan secara keseluruhan pada kekuatan komputasi.



*Gambar 1.1 McCulloch & Pitts, penemu pertama Neural Network*

Hal ini dilanjutkan pada penelitian yang dikerjakan oleh Rosenblatt pada tahun 1950, dimana dia berhasil menemukan sebuah two-layer network, yang disebut sebagai perceptron.

Perceptron memungkinkan untuk pekerjaan klasifikasi pembelajaran tertentu dengan penambahan bobot pada setiap koneksi antar-network.



*Gambar 1.2 Perceptron*

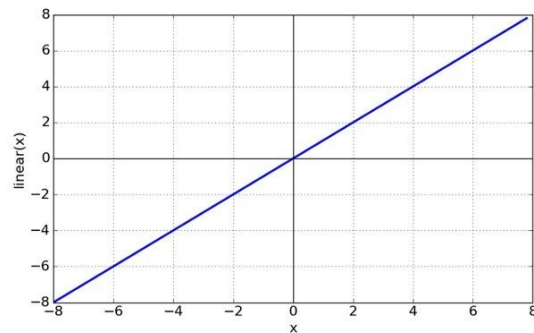
Keberhasilan perceptron dalam pengklasifikasian pola tertentu ini tidak sepenuhnya sempurna, masih ditemukan juga beberapa keterbatasan didalamnya. Perceptron tidak mampu untuk menyelesaikan permasalahan XOR (exclusive-OR). Penilaian terhadap keterbatasan neural network ini membuat penelitian di bidang ini sempat mati selama kurang lebih 15 tahun. Namun demikian, perceptron berhasil menjadi sebuah dasar untuk penelitian-penelitian selanjutnya di bidang neural network. Pengkajian terhadap neural network mulai berkembang lagi selanjutnya di awal tahun 1980-an. Para peneliti banyak menemukan bidang interest baru pada domain ilmu neural network. Penelitian terakhir diantaranya adalah mesin Boltzmann, jaringan Hopfield, model pembelajaran kompetitif, multilayer network, dan teori model resonansi adaptif.

Untuk saat ini, Neural Network sudah dapat diterapkan pada beberapa task, diantaranya classification, recognition, approximation, prediction, clusterization, memory simulation dan banyak task-task berbeda yang lainnya, dimana jumlahnya semakin bertambah seiring berjalannya waktu.

## **b. Activation Function**

Ketika kita membicarakan tentang neural network atau artificial neural network, activation functions merupakan bagian penting karena tugas dari activation function adalah membuat neural network menjadi non-linear. Activation function berfungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan dari weighted sum dari input. Secara umum terdapat 2 jenis activation function, Linear dan Non-Linear Activation function.

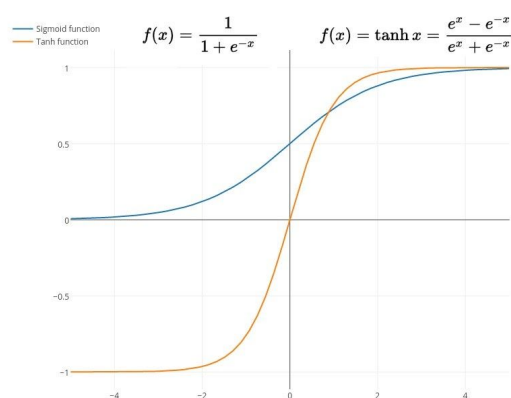
## Linear Function



Gambar 1.3 Linear Function ;  $f(x) = x$

Bisa dikatakan secara “default” activation function dari sebuah neuron adalah Linear. Jika sebuah neuron menggunakan linear function, maka keluaran dari neuron tersebut adalah weighted sum dari input + bias.

## Sigmoid and Tanh Function (Non-Linear)



Gambar 1.4 Sigmoid and Tanh Function

Sigmoid function mempunyai rentang antara 0 hingga 1 sedangkan rentang dari Tanh adalah -1 hingga 1. Kedua fungsi ini biasanya digunakan untuk klasifikasi 2 class atau kelompok data. Namun terdapat kelemahan dari kedua fungsi ini. Sigmoid memiliki bentuk formula sebagai berikut:

$$S(x) = \frac{1}{1 + e^{-x}}.$$

Sigmoid akan menerima angka tunggal dan mengubah nilai  $x$  menjadi sebuah nilai yang memiliki range mulai dari 0 sampai 1. Belakangan ini Sigmoid tidak disukai dan jarang digunakan, sigmoid memiliki kekurangan berupa:

- Sigmoid mematikan gradient, property yang paling tidak diinginkan dari Sigmoid adalah ketika activation dari neuron mengeluarkan nilai yang berada pada ekor 0 atau 1, dimana gradient di wilayah ini hampir nol. Karena itu, jika gradient memiliki nilai yang sangat kecil, Sigmoid akan “mematikan” gradient dan kita sangat tidak menginginkan hal ini terjadi saat melakukan backpropagation.
- Output dari Sigmoid tidak zero-centered. Hal ini berimplikasi pada kedinamisan saat melakukan gradient descent, karena apabila data yang datang ke neuron selalu positif (e.g.  $x > 0$  elementwise pada  $f = w^T x + b$ ) maka gradient pada weights selama backpropagation akan menjadi semua positif atau semua negatif. Hal seperti ini dapat mengganggu proses training, tetapi hal ini tidak separah seperti hal yang ada pada point pertama (mematikan gradient).

Tanh akan mengubah nilai input  $x$  nya menjadi sebuah nilai yang memiliki range mulai dari -1 sampai 1. Sama seperti Sigmoid, Tanh memiliki kekurangan yaitu bisa mematikan gradient, tetapi kelebihanannya adalah output yang dimiliki Tanh merupakan zero-centered. Dalam praktiknya Tanh lebih menjadi pilihan jika dibandingkan dengan Sigmoid. Yang perlu dicatat adalah tanh hanyalah merupakan Sigmoid yang dikembangkan.

$$\tanh(x) = 2\sigma(2x) - 1$$

## ReLU (Non-Linear)

*Gambar 1.4 ReLU Function*

ReLU atau Rectified Linear Unit menjadi salah satu activation function yang populer belakangan ini, Vincent Vanhoucke dalam course deep learningnya di udacity mengatakan bahwa ReLU merupakan activation function favorit para engineer yang malas, kenapa engineer yang malas?

$$f(x) = \max(0, x)$$

Karena ReLU pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila  $x \leq 0$  maka  $x = 0$  dan apabila  $x > 0$  maka  $x = x$ . Ada beberapa pro dan kontra ketika kita menggunakan ReLU:

1. (+) ReLU sangat mempercepat proses konvergensi yang dilakukan dengan stochastic gradient descent jika dibandingkan dengan sigmoid / tanh.
2. (+) Jika kita bandingkan dengan sigmoid/tanh yang memiliki operasi-operasi yang “expensive” (exponentials, etc.), ReLU bisa kita implementasikan hanya dengan membuat pembatas(threshold) pada bilangan nol.
3. (-) Sayangnya, unit ReLU bisa menjadi rapuh pada saat proses training dan bisa membuat unit tersebut “mati”. Sebagai contohnya, kita mungkin bisa menemukan bahwa 40% dari network kita “mati” (neuron yang tidak akan pernah aktif selama proses training) apabila learning rate yang kita inisialisasi terlalu tinggi. Namun apabila kita menginisialisasi learning rate kita secara tepat maka hal seperti ini jarang menjadi masalah.

### c. Learning Algorithm

Jaringan Syaraf Tiruan (JST) merupakan salah satu sistem pemrosesan informasi atau data yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. JST yang berupa susunan sel-sel saraf tiruan (neuron) dibangun berdasarkan prinsip-prinsip organisasi otak manusia. Salah satu metode yang digunakan dalam JST adalah Backpropagation.

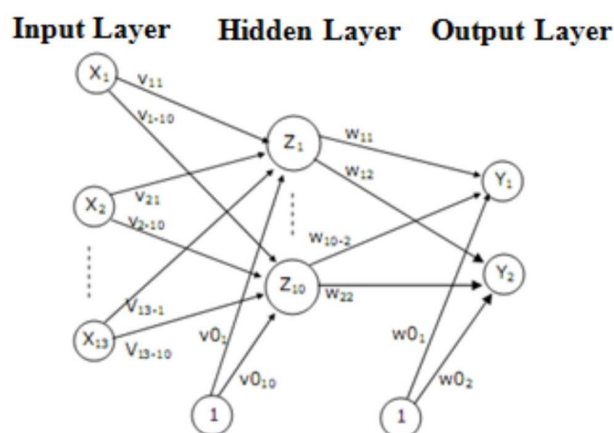
### Pengertian Backpropagation

Backpropagation adalah algoritma pembelajaran untuk memperkecil tingkat error dengan cara menyesuaikan bobotnya berdasarkan perbedaan output dan target yang

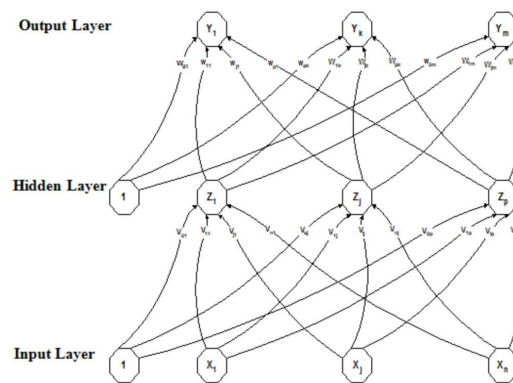
diinginkan. Backpropagation juga merupakan sebuah metode sistematis untuk pelatihan multilayer JST. Backpropagation dikatakan sebagai algoritma pelatihan multilayer karena Backpropagation memiliki tiga layer dalam proses pelatihannya, yaitu input layer, hidden layer dan output layer, dimana backpropagation ini merupakan perkembangan dari single layer network (Jaringan Layer Tunggal) yang memiliki dua layer, yaitu input layer dan output layer. Dengan adanya hidden layer pada backpropagation dapat menyebabkan besarnya tingkat error pada backpropagation lebih kecil dibanding tingkat error pada single layer network. Hal tersebut dikarenakan hidden layer pada backpropagation berfungsi sebagai tempat untuk mengupdate dan menyesuaikan bobot, sehingga didapatkan nilai bobot yang baru yang bisa diarahkan mendekati dengan target output yang diinginkan.

### Arsitektur Backpropagation

Arsitektur algoritma backpropagation terdiri dari tiga layer, yaitu input layer, hidden layer dan output layer. Pada input layer tidak terjadi proses komputasi, namun pada input layer terjadi pengiriman sinyal input  $X$  ke hidden layer. Pada hidden dan output layer terjadi proses komputasi terhadap bobot dan bias dan dihitung pula besarnya output dari hidden dan output layer tersebut berdasarkan fungsi aktivasi tertentu. Dalam algoritma backpropagation ini digunakan fungsi aktivasi sigmoid biner, karena output yang diharapkan bernilai antara 0 sampai 1.



Gambar 1.5 Arsitektur Backpropagation Secara Horizontal



*Gambar 1.6 Arsitektur Backpropagation Secara Horizontal*

Tiga layer backpropagation adalah input layer, hidden layer dan output layer. Pada input layer, inputan divariabelkan dengan  $X_n$ . Pada hidden layer, terdapat bobot ( $V_{ij}$ ) dan bias ( $V_{0j}$ ), serta  $Z$  sebagai data hidden layer. Pada output layer juga demikian, terdapat bobot ( $W_{ij}$ ) dan bias ( $W_{0j}$ ) dengan data output divariabelkan dengan  $Y$ .

### **Algoritma Backpropagation**

Algoritma backpropagation adalah sebuah algoritma untuk memperkecil tingkat error dengan menyesuaikan bobot berdasarkan perbedaan output dan target yang diinginkan. Secara umum algoritmanya terdiri dari tiga langkah utama, yaitu :

1. Pengambilan input
2. Penelusuran error
3. Penyesuaian bobot

Pada pengambilan input, terlebih dahulu dilakukan inisialisasi bobot, kemudian masuk ke dalam algoritma proses backpropagation yang terdiri dari komputasi maju yang bertujuan untuk menelusuri besarnya error dan komputasi balik untuk mengupdate dan menyesuaikan bobot. Dalam mengupdate bobot dapat dilakukan dengan dua cara, yaitu tanpa momentum dan dengan momentum. Namun, yang dijelaskan di bawah ini dalam mengupdate bobotnya dilakukan tanpa memperhatikan besarnya momentum. Dengan demikian dalam metode backpropagation, algoritma yang harus dilakukan adalah inisialisasi bobot, komputasi feed forward dan backpropagation dan inisialisasi kondisi stopping berdasarkan nilai batas error atau jumlah batas epoch. Epoch merupakan rangkaian langkah dalam pembelajaran ANN. Satu epoch diartikan sebagai satu kali pembelajaran ANN.

#### **d. Deep Learning**

Sejarah deep learning dimulai pada tahun 2006, yaitu setelah Geoffrey Hinton mempublikasikan paper yang memperkenalkan salah satu varian neural network yang disebut deep belief nets. Paper ini merupakan awal kemunculan istilah deep learning, untuk membedakan arsitektur neural network konvensional (single layer) dengan arsitektur neural network multi/banyak layer. Ide untuk men-train model jaringan saraf tiruan ini adalah dengan men-train dua layer kemudian tambahkan satu layer di atasnya, kemudian train hanya layer teratas dan begitu seterusnya. Dengan strategi ini kita dapat men-train model jaringan saraf tiruan dengan layer lebih banyak dari model-model sebelumnya. Paper ini merupakan awal populernya istilah deep learning untuk membedakan arsitektur jaringan saraf tiruan dengan banyak layer.

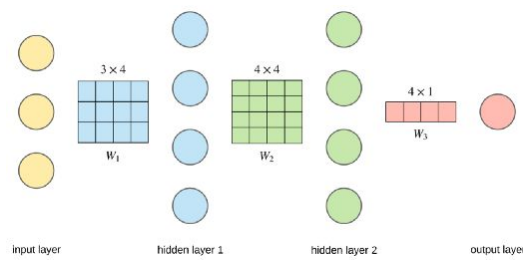
Deep Learning adalah salah satu cabang machine learning yang menggunakan Deep Neural Network untuk menyelesaikan permasalahan pada domain machine learning. Sayangnya ide deep learning masih belum populer di waktu itu, disebabkan algoritma deep learning sangat kompleks, sehingga membutuhkan komputer dengan spesifikasi tinggi yang belum dapat dipenuhi saat itu.

Setelah istilah deep learning populer, deep learning belum menjadi daya tarik yang besar bagi para peneliti karena jaringan saraf tiruan dengan banyak layer memiliki kompleksitas algoritma yang besar, sehingga membutuhkan komputer dengan spesifikasi tinggi, dan tidak efisien secara komputasi saat itu.

Pada tahun 2009, Andrew memperkenalkan penggunaan GPU untuk deep learning melalui paper yang berjudul Large-scale Deep Unsupervised Learning using Graphics Processors. Dengan menggunakan GPU, algoritma deep learning dapat dijalankan lebih cepat dibanding dengan tanpa GPU (hanya menggunakan CPU). Dengan tersedianya hardware yang memadai perkembangan deep learning mulai pesat, dan menghasilkan produk-produk yang dapat kita nikmati saat ini seperti pengenalan wajah, self-driving car, pengenalan suara, dan lain lain.

Bentuk diagram network model deep learning dapat dilihat seperti di bawah ini. Perhatikan bahwa hidden layer hanya digambarkan dua lapis saja. Padahal kenyataannya bisa berjumlah sangat banyak (Misal mencapai ribuan buah).





*Gambar 1.7 Diagram Network Model Deep Learning*

Deep learning sudah dikembangkan ke berbagai model atau arsitektur yang berbeda-beda. Beberapa model atau arsitektur untuk deep learning : Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Haar Hascade, Deep Belief Networks (DBN) Dan You Only Look Once (YOLO).

#### **e. Convolutional Neural Network**

Convolutional Neural Networks (ConvNets) merupakan special case dari artificial neural networks (ANN) yang saat ini diklaim sebagai model terbaik untuk memecahkan masalah object recognition dan detection. Trend terakhir di bidang Computer Vision, kita akan sangat mudah mencari paper yang di judulnya terdapat kata “Deep Learning” atau “Convolutional”. Model ini juga merupakan model favorit di ranah industri. Perusahaan-perusahaan besar seperti Google, Facebook, Microsoft, dan Baidu membangun pusat riset khusus di bidang ini.

Di ranah machine learning (atau neural networks pada khususnya), ConvNets sebenarnya bukan barang baru. Perkembangan awal ConvNets dapat ditelusuri dari [Fukushima 1980] yang merancang sebuah unsupervised artificial networks bernama Neocognitron, terinspirasi dari sistem visual biologis yang dirancang oleh Hubel dan Wiesel — penemuan Hubel dan Wiesel pernah meraih Nobel Prize Winner di bidang fisiologi pada tahun 1981. LeCun et al. 1989 mengekstensi model tersebut di berbagai sisi dan mengaplikasikannya pada konteks supervised learning untuk memecahkan masalah handwritten digit recognition. Istilah “convolutional network networks” sendiri diperkenalkan di LeCun et al. 1998. Model arsitekturnya dikenal dengan nama LeNet.

Di akhir 90-an hingga pertengahan tahun 2000-an, neural networks sempat “nyaris dilupakan” dikarenakan muncul berbagai algoritma (e.g., Support Vector Machines,

AdaBoost) yang dapat dieksekusi lebih cepat dengan performa yang lebih baik pada waktu itu. Neural networks kembali mendapatkan perhatian ketika Deep Belief Networks (DBN) [Hinton et al. 2006] membuat terobosan dengan menjadi model handwritten digit recognition yang paling akurat, yang pada akhirnya memunculkan istilah deep learning.

Sebuah neural network disebut dengan “deep” apabila memiliki layer hirarkis yang banyak/bertumpuk-tumpuk. Perlu diketahui bahwa sebelum DBN diperkenalkan, sudah menjadi semacam best practice bahwa model standard neural networks yang efektif hanya memiliki 2 buah layer. Menambah jumlah layer ‘dipercayai’ tidak meningkatkan atau bahkan memperburuk performa. Dapat dikatakan bahwa DBN merupakan arsitektur “deep” / lebih dari 2 layer yang pertama yang mampu menghasilkan breakthrough.

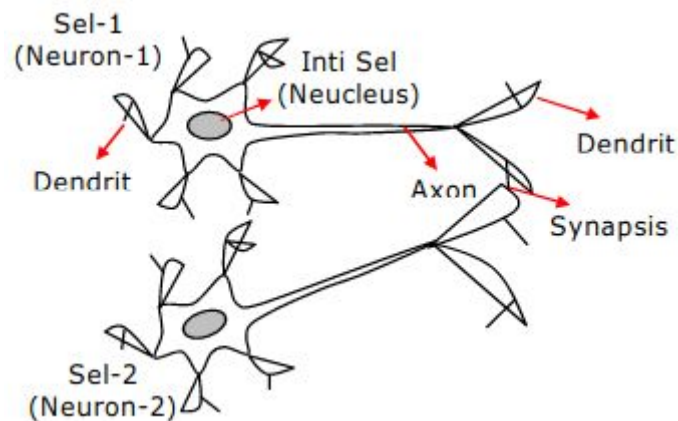
Berbeda dengan model-model neural networks seperti biasanya (termasuk ConvNets) yang hanya memiliki 1 tahap training (supervised back-propagation), DBN memiliki 2 tahap training terpisah: 1) unsupervised pre-training (dengan menggunakan restricted Boltzmann Machines) dan 2) supervised training (back-propagation). Pada saat itu, tahap unsupervised pre-training inilah yang dipercayai sebagai alasan mengapa arsitektur yang “deep” dapat bekerja dengan baik.

Pada tahun 2012 muncul terobosan yang baru lagi pada deep learning. ConvNets dengan arsitektur tertentu yang dipadukan dengan berbagai trik (e.g., drop out regularization, pemanfaatan Rectified Linear Unit (ReLU) sebagai fungsi aktivasi, data augmentation) mampu mencapai terobosan pada large scale image classification (ImageNet) yang memiliki 1000 kategori objek dan ~1 juta gambar, melebihi performa manusia. Model ini dikenal dengan nama AlexNet. Yang cukup mengagetkan adalah proses training-nya kembali ke jaman dahulu: hanya menggunakan 1 fase saja, yaitu supervised backpropagation, sehingga menyebabkan banyak orang tidak mempercayai lagi pengaruh dari unsupervised pre-training.

## **2. Neural Network**

Neural network atau jaringan saraf tiruan merupakan representasi buatan dari adanya otak manusia yang mencoba dalam mensimulasikan proses pembelajaran seperti otak manusia. Representasi buatan tersebut adalah jaringan syaraf yang diimplementasikan dengan menggunakan program komputer serta mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Adapun meniru dari otak manusia karena dalam

otak manusia berisi banyak sel saraf yang bertugas untuk memproses informasi. Setiap sel bekerja seperti suatu prosesor sederhana, masing-masing sel tersebut saling berinteraksi supaya mendukung kemampuan kerja otak manusia.

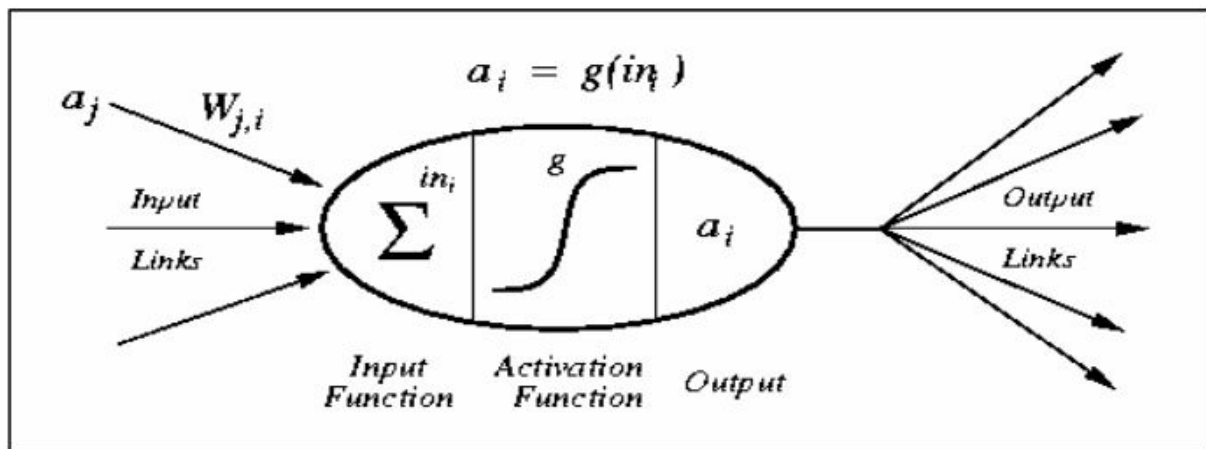


*Gambar 2.1 Susunan Saraf Otak Manusia*

#### **a. Neuron**

Dalam otak manusia terdapat bagian yang penting salah satunya yaitu sel saraf atau neuron merupakan satuan unit pemroses terkecil pada otak dan memiliki satu inti sel. Inti sel tersebut yang akan bertukar untuk melakukan pemrosesan informasi. Informasi yang masuk akan diterima oleh dendrit (berfungsi penyampai sinyal ke neuron) dan setelah menerima informasi, dendrit tersebut juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi dari hasil tersebut akan menjadi masukan bagi neuron lain antara dendrit pada kedua sel tersebut dipertemukan dengan synapsis.

Informasi yang dikirimkan antara neuron tersebut berupa rangsangan yang melewati dendrit. Informasi yang masuk dan diterima oleh dendrit akan dijumlahkan dan dikirim melalui axon ke dalam dendrit akhir yang bersentuhan dengan dendrit dari neuron yang lain. Informasi tersebut selanjutnya akan diterima oleh neuron lain, jika memenuhi batasan nilai ambang (*threshold*) dan neuron tersebut dikatakan teraktivasi. Hubungan antar neuron sebelumnya terjadi secara adaptif sehingga struktur hubungan tersebut terjadi secara dinamis. Oleh karena itu otak manusia memiliki kemampuan untuk belajar dengan melakukan adaptasi. Pada jaringan saraf tiruan, tiruan neuron meniru sebagaimana neuron aslinya bekerja, tiruan neuron tersebut diilustrasikan sebagai berikut:



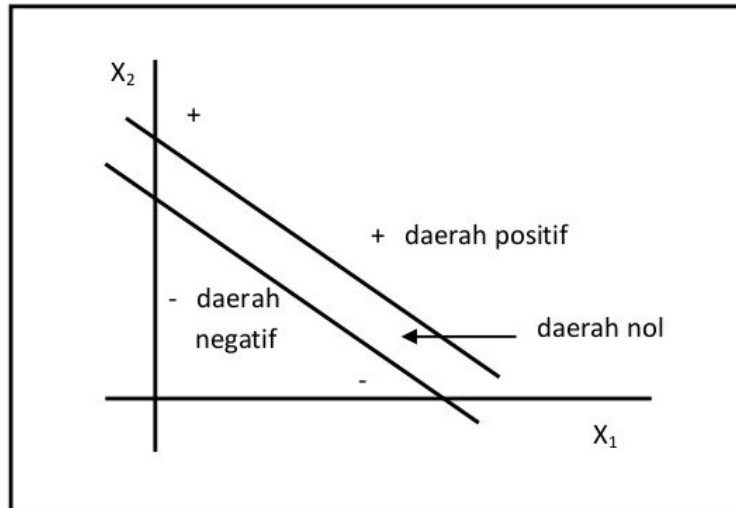
*Gambar 2.2 Model Tiruan Neuron*

Dari gambar di atas terlihat sinyal masukan dari variabel  $a$  yang kemudian dikalikan dengan masing-masing penimbang yang bersesuaian atau variabel  $w$ . Selanjutnya dilakukan sebuah penjumlahan dari seluruh hasil perkalian tersebut dan menghasilkan keluaran ke dalam fungsi aktivasi untuk mendapatkan tingkatan derajat sinyal keluaran. Itulah tiruan neuron dalam jaringan saraf tiruan, kumpulan dari neuron tersebut dibuat menjadi sebuah jaringan yang berfungsi sebagai alat komputasi. Setiap neuron dan struktur jaringan digunakan untuk menyelesaikan suatu permasalahan secara berbeda-beda.

## **b. Perceptron**

Bagian jaringan saraf lainnya yaitu perceptron yang digunakan untuk mengklasifikasikan suatu tipe pola tertentu dengan pemisahan secara linear. Algoritma yang digunakan oleh perceptron ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Perceptron ini juga merupakan metode pembelajaran dengan pengawasan dalam sistem jaringan saraf dan merancang jaringan neuron dengan banyaknya spesifikasi yang akan diidentifikasi.

Perceptron pada dasarnya yang memiliki satu lapisan maka mempunyai bobot yang dapat diatur dan mempunyai suatu nilai threshold. Nilai threshold ( $\theta$ ) pada fungsi aktivasi adalah non negatif. Fungsi aktivasi ini dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif.



Gambar 2.3 Pembatasan Linear dengan perceptron

Pada gambar di atas terlihat garis pemisah antara daerah positif dan daerah nol memiliki pertidaksamaan :

$$W_1X_1 + W_2X_2 + \dots + W_nX_n + b > \Theta$$

Sementara garis pemisah antara daerah negatif dengan daerah nol memiliki pertidaksamaan :

$$W_1X_1 + W_2X_2 + \dots + W_nX_n + b < -\Theta$$

Berikut algoritma pada perceptron :

1. Inisialisasi semua bobot dan bias untuk mengatur semua bobot dan bias sama dengan nol serta atur learning rate :  $\alpha$  ( $0 < \alpha \leq 1$ ).
2. Selama kondisi berhenti dan bernilai false, maka akan melakukan langkah berikut:
  - a. Setiap pasangan pembelajaran s-t mengerjakan
    - i. Set input dengan nilai sama dengan vektor input, seperti berikut:

$$X_i = S_i;$$

- ii. Hitung respon untuk unit output, seperti berikut:

$$y_{in} = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1, & \text{jika } y_{in} > \theta \\ 0, & \text{jika } -\theta \leq y_{in} \leq \theta \\ -1, & \text{jika } y_{in} < -\theta \end{cases}$$

iii. Perbaiki bobot dan bias jika terjadi error, seperti berikut:

Jika  $y \neq t$  maka:

$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha * t * x_i$$

$$b(\text{baru}) = b(\text{lama}) + \alpha * t$$

jika tidak, maka:

$$w_i(\text{baru}) = w_i(\text{lama})$$

$$b(\text{baru}) = b(\text{lama})$$

- b. Tes kondisi jika berhenti, ketika tidak terjadi perubahan bobot pada setiap pasangan pembelajaran s-t maka kondisi berhenti TRUE, namun jika masih terjadi perubahan maka kondisi berhenti FALSE.

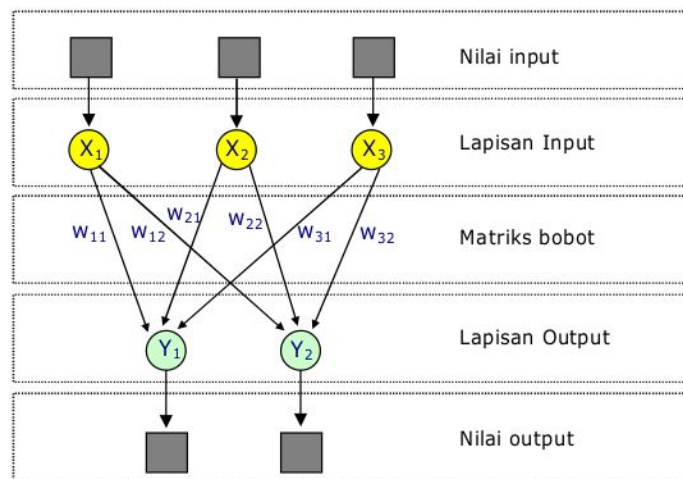
Algoritma di atas dapat digunakan baik untuk input biner maupun bipolar. Pada algoritma ini bobot yang diperbaiki hanyalah bobot yang berhubungan dengan input yang aktif dan bobot yang tidak menghasilkan nilai  $y$  maka itu yang benar.

### c. Arsitektur Neural Network

Dalam jaringan saraf tiruan terdapat juga arsitekturnya yang berguna untuk mengelompokkan ke dalam lapisan-lapisan, setiap lapisan tersebut terdapat neuron-neuron yang saling terhubung. Diantara arsitektur tersebut sebagai berikut:

#### 1. Single Layer Net

Pada single layer net hanya memiliki satu lapisan dengan bobot yang saling terhubung. Jaringan ini hanya menerima input kemudian langsung mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.

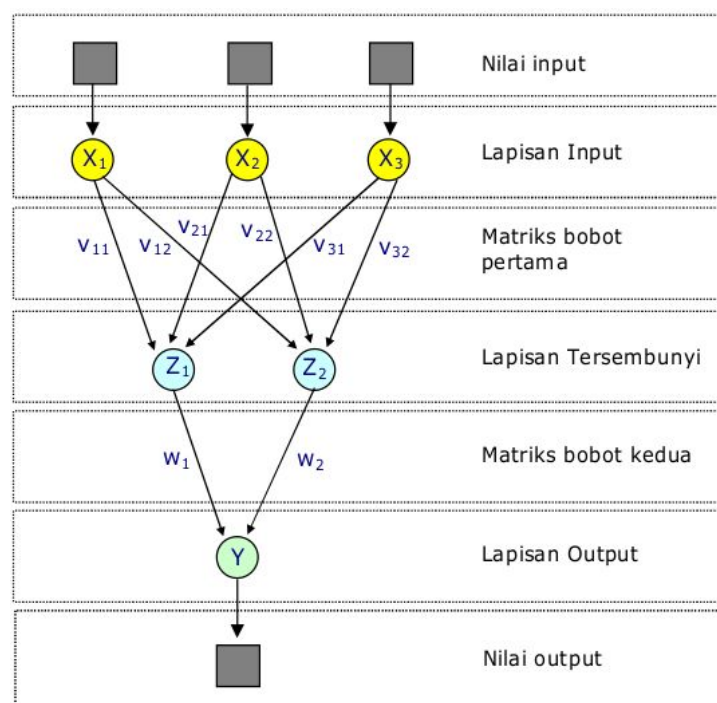


Gambar 2.4 Arsitektur Single Layer

Contoh pada gambar di atas lapisan input memiliki 3 neuron, yaitu  $X_1$ ,  $X_2$  dan  $X_3$ . Sedangkan pada lapisan output memiliki 2 neuron yaitu  $Y_1$  dan  $Y_2$ . Neuron pada kedua lapisan saling berhubungan. Besarnya hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian dan semua unit input akan dihubungkan dengan setiap unit output.

## 2. Multi Layer Net

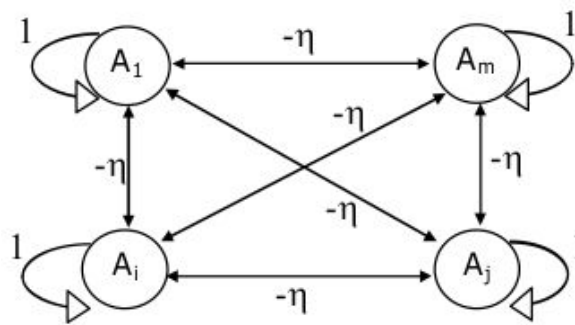
Multi layer net memiliki 1 atau lebih lapisan yang terletak diantara lapisan input dan lapisan output atau lapisan tersembunyi). Secara umum terdapat lapisan bobot yang terletak antara 2 lapisan yang bersebelahan serta dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal, karena melakukan pembelajaran yang lebih rumit. Akan tetapi dalam banyak kasus pembelajaran ini lebih sukses dalam menyelesaikan masalah.



*Gambar 2.5 Arsitektur Multi Layer*

### 3. Competitive Layer Net

Competitive layer net umumnya memiliki hubungan antar neuron pada lapisan kompetitif ini dan tidak diperlihatkan pada diagram arsitektur. Pada gambar di bawah menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot  $-\eta$ .



Gambar 2.6 Arsitektur Competitive Layer

#### d. Fungsi Aktivasi

Jaringan saraf tiruan terdapat fungsi untuk menentukan keluaran yang disebut fungsi aktivasi. Fungsi aktivasi yang digunakan diantaranya:

1. Fungsi Undak Biner Hard Limit, berfungsi mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output biner 0 atau 1.
2. Fungsi Undak Biner Threshold, berfungsi mengkonversikan input dengan menggunakan nilai ambang dengan output biner 0 atau 1.
3. Fungsi Bipolar Symetric Hard Limit, berfungsi mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output 1, 0, atau -1.
4. Fungsi Bipolar Threshold, berfungsi mengkonversikan input dengan menggunakan nilai ambang dengan output 1, 0, atau -1.
5. Fungsi Linear, fungsi yang memiliki nilai input dan output yang sama.
6. Fungsi Saturating Linear, fungsi yang jika input kurang dari -0,5 maka output bernilai 0, akan bernilai 1 jika input lebih dari 0,5 dan jika input diantara -0,5 sampai 0,5 maka bernilai sama dengan input hanya saja ditambah 0,5.
7. Fungsi Symetric Saturating Linear, fungsi yang jika input kurang dari -1 maka bernilai -1, akan bernilai 1 jika input lebih dari 1 dan jika input diantara -1 sampai 1 maka bernilai sama dengan input.

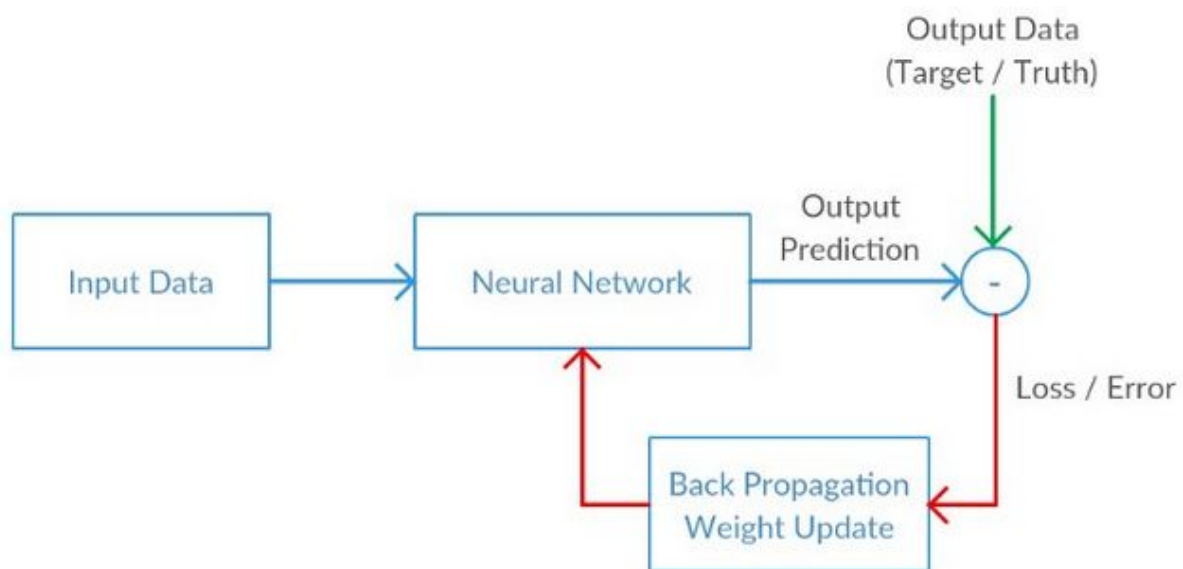


8. Fungsi Sigmoid Biner, berfungsi untuk jaringan saraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1.
9. Fungsi Sigmoid Bipolar, berfungsi untuk jaringan saraf yang membutuhkan nilai output yang terletak pada interval 1 sampai -1.

#### e. Loss Function

Loss function atau fungsi rugi merupakan perbedaan antara hasil keluaran dan prediksi keluaran. Fungsi utama dari loss function untuk mengukur seberapa bagus performa dari neural network kita dalam melakukan prediksi terhadap target.

$$Loss = (Target - Prediction)^2$$



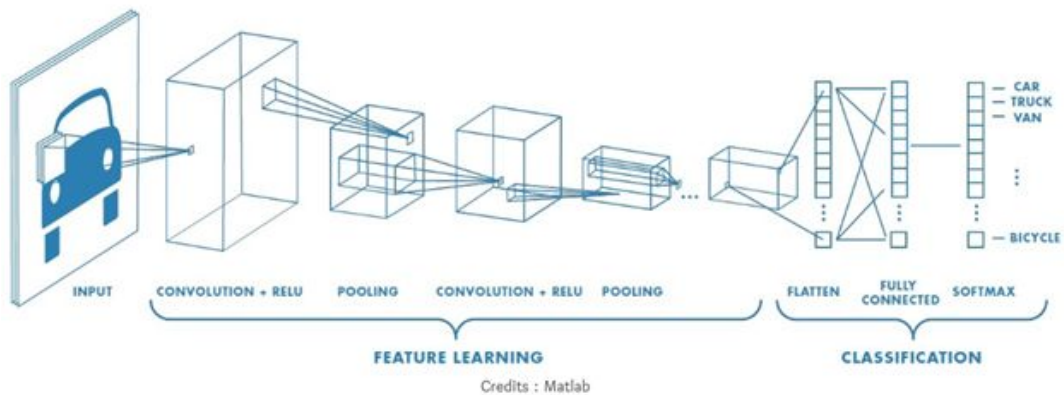
*Gambar 2.7 Neural Network Training*

### 3. Convolutional Neural Network

Convolutional neural network adalah salah satu metode yang digunakan pada machine learning yang berfungsi untuk mengolah data 2 dimensi atau gambar. Untuk mengenali atau mendeteksi object pada gambar salah satu metodenya menggunakan CNN. Pemrosesan image pada cnn sendiri terdiri dari beberapa layer. Layer tersebut akan mengekstraksi image yang nanti nya bisa diolah untuk penerapan machine learning.

### a. Arsitektur Dan Layer Pada CNN

CNN sendiri mempunyai arsitektur, yang pada arsitektur nya terdiri dari beberapa layer. Penerapan arsitektur dan layer yang berada di dalamnya adalah sebagai berikut:



Gambar 3.1 Arsitektur dan Layer CNN

### Feature Learning

Layer yang berada pada feature learning digunakan untuk mengolah input yang mempunyai bentuk angka dalam suatu vektor agar diterjemahkan menjadi suatu fitur. Layer yang berada pada feature learning adalah convolutional layer dan pooling layer

#### 1. Convolutional layer

Digunakan untuk mengolah output agar dapat dihitung. Output nya sendiri terdapat pada neuron yang terkoneksi dengan input pada daerah lokal. Bobot dan wilayah kecil yang terkoneksi pada volume input dihitung produk titik nya.

#### 2. Rectified Linear Unit

Digunakan untuk menerapkan fungsi aktivasi elemen pada saat diambang batas 0 dengan rumus  $f(x) = \max(0, x)$  yang nantinya berfungsi agar vanishing gradient menghilang.

#### 3. Pooling layer

Digunakan untuk downsampling yang berfungsi agar feature map dapat mengurangi dimensinya supaya komputasi dapat dipercepat karena semakin sedikit parameter yang diupdate dan menyelesaikan masalah overfitting.

## **Classification**

Layer ini digunakan untuk melakukan klasifikasi pada neuron yang dihasilkan dari ekstraksi fitur sebelumnya. Klasifikasi tersebut terdiri dari:

### **1. Flatten**

Melakukan reshape atau membentuk ulang feature map agar diubah menjadi vector yang nantinya digunakan sebagai input agar bisa diolah di fully connected layer.

### **2. Fully-Connected**

Digunakan untuk menghitung skor kelas pada klasifikasi nya. Angka yang terdapat pada volume akan terkoneksi dengan neuron pada layer ini.

### **3. Softmax**

Probabilitas pada setiap kelas target dihitung dengan tujuan membantu input yang diberikan agar dapat ditentukan kelas targetnya

## **b. Training**

Proses training pada CNN adalah sebagai berikut:

### **1. Mengumpulkan dataset**

Dataset adalah data yang digunakan untuk pemrosesan CNN. Data yang digunakan sebagai dataset akan dibagi menjadi training, validation, dan testing. Data yang digunakan pada training akan digunakan untuk melatih model CNN yang diterapkan, Lalu data pada validation digunakan untuk memvalidasi model CNN yang telah ditraining. Model hasil dari training akan dilakukan testing menggunakan data yang digunakan khusus untuk testing untuk melihat keakuratan model.

### **2. Feedforward input to output**

Pada proses ini dataset yang digunakan pada training akan dilakukan pengklasifikasian, dimana inputan tersebut akan berjalan dan melewati output untuk mendapatkan hasil pengklasifikasian.

### **3. Backpropagation**

Digunakan untuk melakukan training pada model CNN yang digunakan. Proses pada algoritma ini adalah melihat error pada output untuk diubah bobot pada nilai nya secara mundur.

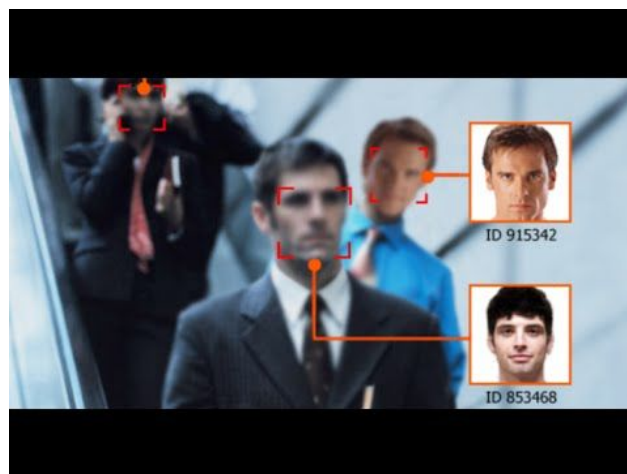
#### 4. Stochastic Gradient Descent

Digunakan untuk meminimalkan error ketika melakukan proses training sehingga menghasilkan nilai yang lebih optimal. Proses ini terjadi pada backpropagation. Dalam proses ini juga terjadi beberapa bagian, seperti epoch dimana kita melihat bagaimana seluruh data ditraining pada cnn dalam satu iterasi. Lalu batch size dimana data tersebut dibagi bagi menjadi beberapa bagian untuk disebar ke CNN tujuan nya sendiri adalah agar data bisa lebih optimal. Lalu iterasi yaitu berapa kali iterasi yang diperlukan untuk melakukan training pada CNN. Learning Rate yaitu hasil di tiap training nya menghasilkan seberapa dekat dengan output yang diinginkan

#### 4. Aplikasi

Berikut beberapa penerapan algoritma Convolutional Neural Network dalam bentuk aplikasi yaitu:

- a. Closed Circuit Television atau sering disebut CCTV merupakan penerapan dari CNN dengan memasukkan dataset ke dalam database yang berupa picture-picture kemudian dilakukan training hingga pada saat implementasi, CCTV dapat merekam video dan melabelkan setiap objek yang sudah berada dalam dataset dan membedakan dengan objek yang tidak ada dalam dataset.



*Gambar 4.1 CCTV Face Recognition*

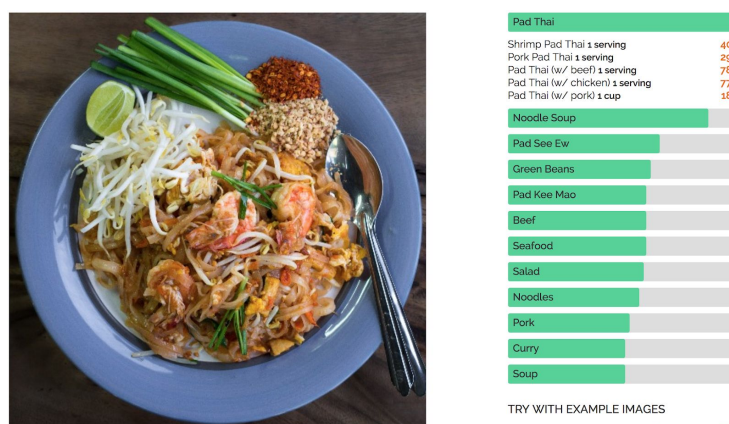
- b. Pendeteksi Patah Tulang dengan melakukan pemindaian pada tulang dengan atau tanpa daging sekalipun. Citra yang digunakan biasanya menggunakan probe US yang

kemudian hasil pemindaian tersebut dijadikan bentuk video dengan banyak citra yang telah dipindai dari kondisi tulang tersebut. Setelah mendapatkan hasilnya maka diperlukan waktu untuk komputasi hingga mendapatkan hasil apakah tulang tersebut patah atau tidak.



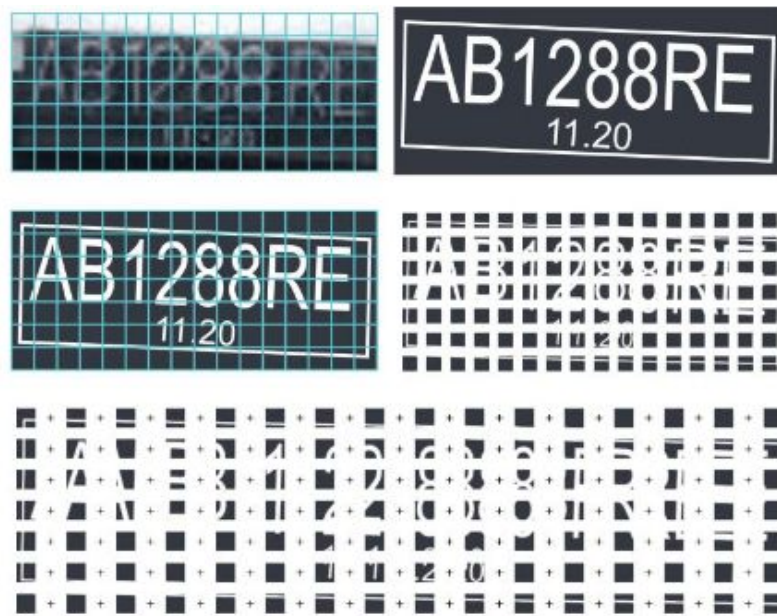
*Gambar 4.2 Pemindaian Tulang*

- c. Prediksi Citra Makanan dengan melakukan proses preprocessing terlebih dahulu dan kemudian dilakukan training dengan proses feedforward serta backpropagation. Hasil akhirnya yaitu proses testing setelah mentraining data secukupnya, maka akan didapatkan akurasi dari sebuah makanan yang diambil citranya dengan keluaran bisa berupa jenis makanan, kalori, gizi, dan sebagainya.



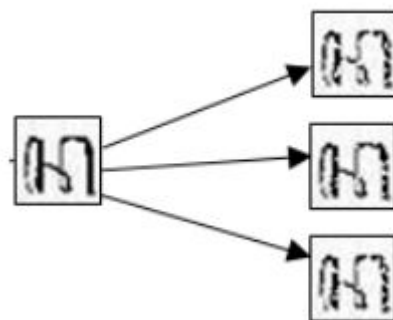
*Gambar 4.3 Food Recognition*

- d. Pendeteksi Tanda Nomor Kendaraan Bermotor yang digunakan pada media streaming terbagi menjadi beberapa layer yaitu layer input, layer convolution, layer activation, layer pooling dan fully connected layer. Berdasarkan layer-layer berikut dengan menggunakan algoritma CNN maka dapat memiliki tingkat akurasi yang cukup besar dan tidak terlepas dari beragamnya dataset yang mempengaruhi.



*Gambar 4.4 Pendeteksi Tanda Nomor Kendaraan Bermotor*

- e. Pengenalan Teks salah satunya pengenalan aksara jawa yaitu dengan melakukan pemilihan data dari hasil segmentasi sehingga menjadi dataset untuk ditraining hingga diberi label. Hal tersebut sangat mempengaruhi tingkat akurasi dari hasil pengujian yang akan dilakukan, terlebih lagi dengan menggunakan algoritma CNN.



*Gambar 4.5 Pengenalan Aksara Jawa*



- f. Object Tracking in Games juga merupakan contoh penerapan algoritma CNN dengan melakukan tracking pada objek-objek dalam game tersebut sehingga objek tersebut dapat terdeteksi melalui sebuah citra yang telah dilakukan komputasi sebelumnya.



*Gambar 4.6 Object Tracking in Games*

## 5. Referensi

- <https://socs.binus.ac.id/2012/07/26/konsep-neural-network/>
- <https://medium.com/@haiqalmuhamadalfarisi/mengenal-perbedaan-artificial-intelligence-machine-learning-neural-network-deep-learning-part-bef034a145bd#>
- <https://www.codepolitan.com/mengenal-teknologi-deep-learning-dan-sejarahnya-59aea44b5f64>
- <http://yuliana.lecturer.pens.ac.id/Kecerdasan%20Buatan/Buku/Bab%208%20Jaringan%20Syaraf%20Tiruan.pdf>
- [http://share.its.ac.id/pluginfile.php/14163/mod\\_resource/content/1/jst1.pdf](http://share.its.ac.id/pluginfile.php/14163/mod_resource/content/1/jst1.pdf)
- <https://medium.com/@opam22/menilik-activation-functions-7710177a54c9>
- <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>
- <https://jurnal.umk.ac.id/index.php/simet/article/view/1737/0>

- <https://medium.com/@nadhifasofia/1-convolutional-neural-network-convolutional-neural-network-merupakan-salah-satu-metode-machine-28189e17335b>
- [https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4#:~:text=Convolutional%20Neural%20Network%20\(CNN\)%20adalah,persepsi%20visual%20seperti%20contoh%20diatas.](https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4#:~:text=Convolutional%20Neural%20Network%20(CNN)%20adalah,persepsi%20visual%20seperti%20contoh%20diatas.)
- <https://www.google.com/amp/s/ghifar.wordpress.com/2015/07/21/deep-convolutional-neural-networks-part-1/amp/>
- <https://www.google.com/amp/s/novikaginanto.wordpress.com/2012/11/14/backpropagation/amp/>
- <http://andrey.web.id/blog/membangun-jaringan-saraf-tiruan-dari-awal-pengenalan/>
- <https://medium.com/@samuelsena/pengenalan-deep-learning-part-3-backpropagation-algorithm-720be9a5fbb8>
- <http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/download/585/357>
- <http://ejnteti.jteti.ugm.ac.id/index.php/JNTETI/article/download/491/409>
- [https://www.researchgate.net/profile/Rizky\\_Dwi\\_Novyantika/publication/323511066\\_DETEKSI\\_TANDA\\_NOMOR\\_KENDARAAN\\_BERMOTOR\\_PADA\\_MEDIA\\_STREAMING\\_DENGAN\\_ALGORITMA\\_CONVOLUTIONAL\\_NEURAL\\_NETWORK\\_MENGGUNAKAN\\_TENSORFLOW/links/5a98dd21a6fdccceff0d43e1/DETEKSI-TANDA-NOMOR-KENDARAAN-BERMOTOR-PADA-MEDIA-STREAMING-DENGAN-ALGORITMA-CONVOLUTIONAL-NEURAL-NETWORK-MENGGUNAKAN-TENSORFLOW.pdf?origin=publication\\_detail](https://www.researchgate.net/profile/Rizky_Dwi_Novyantika/publication/323511066_DETEKSI_TANDA_NOMOR_KENDARAAN_BERMOTOR_PADA_MEDIA_STREAMING_DENGAN_ALGORITMA_CONVOLUTIONAL_NEURAL_NETWORK_MENGGUNAKAN_TENSORFLOW/links/5a98dd21a6fdccceff0d43e1/DETEKSI-TANDA-NOMOR-KENDARAAN-BERMOTOR-PADA-MEDIA-STREAMING-DENGAN-ALGORITMA-CONVOLUTIONAL-NEURAL-NETWORK-MENGGUNAKAN-TENSORFLOW.pdf?origin=publication_detail)
- <https://www.youtube.com/watch?v=uPwtmT4eklk>
- <https://jurnal.undhirabali.ac.id/index.php/jutik/article/download/1001/pdf>
- <https://medium.com/@pkuhar/re-comparison-of-image-recognition-apis-on-food-images-976979793ae0>
- <http://publication.petra.ac.id/index.php/teknik-informatika/article/download/8075/7285>
- <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=3192&context=theses>