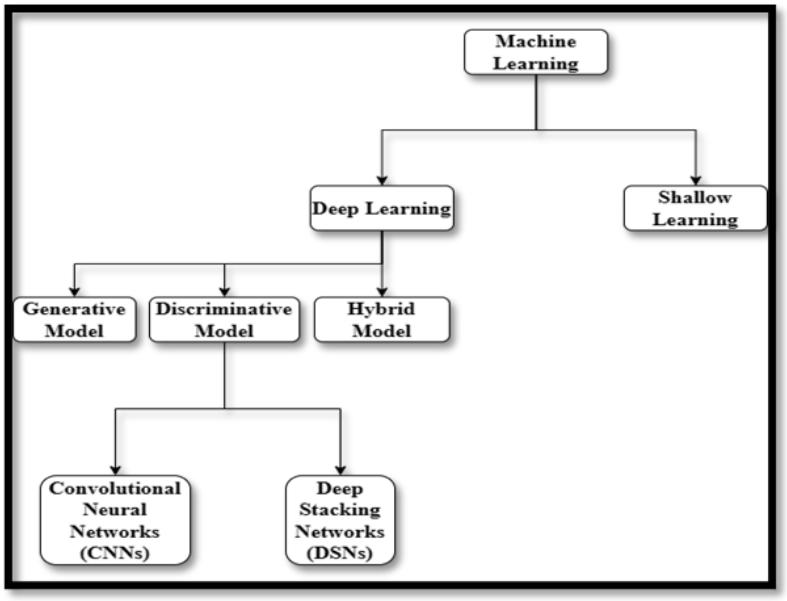
BAB II
TINJAUAN PUSTAKA

2.1 *Machine Learning*

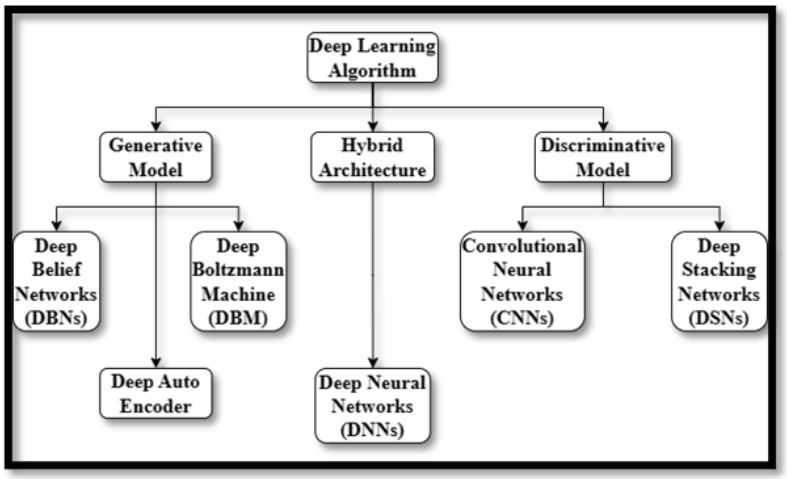
Machine Learning merupakan salah satu bagian dari teknologi kecerdasan buatan guna membantu mesin bagaimana cara belajar. Pembelajaran mesin ini didasarkan pada sebuah algoritma yang membuat suatu komputer untuk belajar dan melakukan pekerjaannya tanpa perintah dari pengguna [8]. *Machine Learning* terdiri dari dua jenis, yakni *Deep Learning* dan *Shallow Learning*. *Shallow learning* merupakan pembelajaran dalam *Machine Learning* yang merujuk pada penggunaan teknik pembelajaran mesin yang beroperasi pada fitur yang telah ditentukan dan dirancang secara manual serta algoritma yang relatif sederhana. Pada umumnya, *Shallow Learning* digunakan dalam penanganan data terstruktur di industri konstruksi, seperti indikator peringatan dini pemantauan keselamatan [9]. Pada gambar 2.1 merupakan diagram *Machine Learning*.

2.2 *Deep Learning*

Deep Learning merupakan sekumpulan algoritma pembelajaran mesin yang menggunakan banyak lapisan sesuai dengan tingkat abstraksi yang berbeda pada setiap levelnya [10]. Deep learning banyak digunakan untuk aplikasi sintesis suara, pemrosesan gambar, pengenalan tulisan tangan, pendeteksi objek, analisis prediksi, dan pengambilan keputusan. *Deep Learning* terbagi menjadi tiga jenis, yaitu *Generative Model*, *Hybrid Architecture*, dan *Discriminative Model*. Pada gambar 2.2 adalah merupakan bagan dari jenis - jenis metode *deep learning Deep Learning*.



Gambar 2.1. Diagram *Machine Learning*

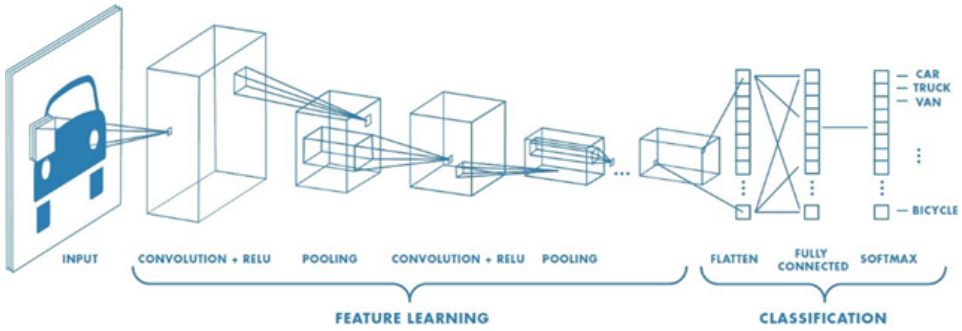


Gambar 2.2. Diagram *Deep Learning*

2.3 *Convolutional Neural Network*

Convolutional Neural Network merupakan jenis *Deep Neural Network* yang menggunakan data dua dimensi dengan kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [11]. Konvolusi adalah operasi matematis yang memiliki fungsi untuk mengekstrak fitur penting dari citra pada gambar secara berulang-ulang. *Output* yang diterima dari proses konvolusi adalah *Feature map*, yang merupakan representasi matematis dari fitur-fitur dalam citra. *Convolutional Neural Network* memiliki beberapa layer yang digunakan untuk melakukan proses *filter* di

setiap prosesnya. Proses yang dijalankan disebut proses *training*. Pada proses *training* terdapat 3 tahapan, yaitu tahap *Convolutional layer*, *Pooling layer*, dan *Fully connected layer*. Pada gambar 2.3 menjelaskan proses tahapan dari CNN.



Gambar 2.3. Proses *Convolutional Neural Network*

2.3.1 *Convolutional Layer*

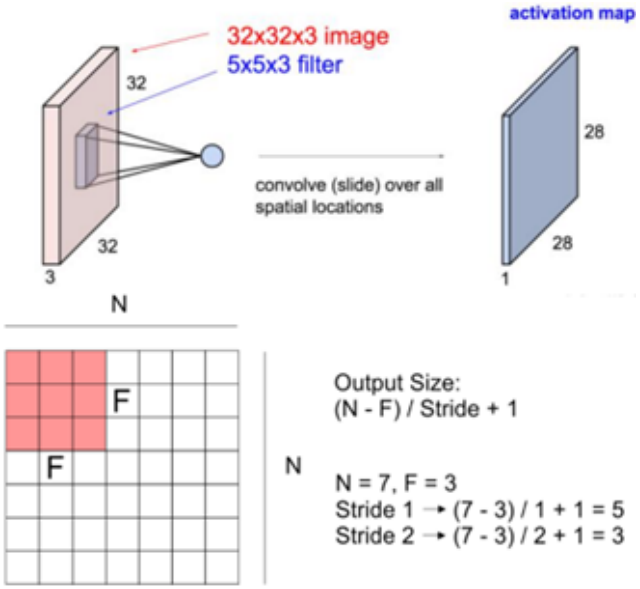
Convolutional Layer adalah lapisan dalam arsitektur *Convolutional Neural Network* yang melakukan proses konvolusi pada suatu citra/gambar. Seluruh data yang mencapai lapisan konvolusi akan diolah melalui proses konvolusi. Lapisan akan mengkonversi setiap *filter* ke seluruh bagian data masukan dan menghasikan sebuah *activation map* atau *feature map* 2D [12]. Persamaan rumus dari *Feature Map* 2D dapat dituliskan sebagai berikut :

$$nout = \left(\frac{nin - k + 2p}{s}\right) + 1 \tag{2.1}$$

Dimana:

nout	= Ukuran *feature map*
nin	= Ukuran matriks masukan
k	= Ukuran matriks filter
p	= Ukuran *padding*
s	= *stride*

Filter yang terdapat pada *Convolutional Layer* memiliki panjang, tinggi (*pixels*) dan tebal sesuai dengan *channel data* masukan. Setiap *filter* akan mengalami pergeseran dan operasi "dot" antara data masukan dan nilai dari *filter*. *Convolutional Layer* secara signifikan mengalami kompleksitas model melalui optimalisasi outputnya. Hal ini dioptimalkan melalui tiga parameter, *depth*, *stride* dan pengaturan *zero padding*. Pada gambar 2.4 menjelaskan tentang bagan *Convolutional Layer*.



Gambar 2.4. *Convolutional Layer*

Adapun rumus dari operasi konvolusi dapat dituliskan sebagai berikut :

$$FM[i]_{j,k} = \left(\sum_{m} \sum_{n} N_{[j-m,k-n]} F_{[m,n]} \right) + bF \tag{2.2}$$

Dimana:

$FM[i]$	= Matriks *feature maps* ke - i
j,k	= Posisi *pixel* pada maktriks citra input
m,n	= Posisi *pixel* pada maktriks *filter* konvolusi
N	= Matriks citra masukan
F	= Matriks *filter* konvolusi
bF	= Nilai bias pada *filter*

2.3.2 *Pooling Layer*

Pooling Layer adalah lapisan yang menggunakan fungsi dengan *Feature Map* sebagai masukan dan mengolahnya dengan berbagai macam operasi statistik berdasarkan nilai piksel terdekat [13]. Pada model CNN, Setiap pergeseran akan ditentukan oleh berapa banyak stride yang akan digeser pada seluruh area *feature map* atau *activation map*. Dua jenis *Pooling Layer* yang umum digunakan adalah *Max Pooling* dan *Average Pooling*. Sebagai contoh, Ketika menggunakan *Max Pooling* dengan ukuran 2×2 dan *Stride* 2, maka nilai yang didapatkan setiap pergeseran *filter*nya adalah nilai yang terbesar pada area 2×2 tersebut. Sedangkan *Average Pooling* akan memindai rata - ratanya. Pada gambar 2.5 menjelaskan tentang bagan *Pooling Layer*.

Gambar 2.5. *Pooling Layer*

Persamaan rumus resolusi *feature map max pooling* dapat dituliskan sebagai berikut

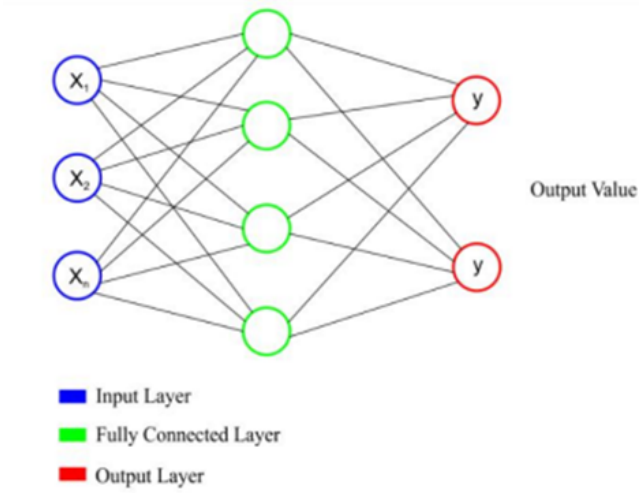
$$a_i = \max_{N \times N} \quad a_j^{n \times n} \mu(n,n) \tag{2.3}$$

Dimana:

a_i = *value* dari *pooling map*
a_j = operasi penjumlahan
$\mu(n,n)$ = *windows function*

2.3.3 *Fully Connected Layer*

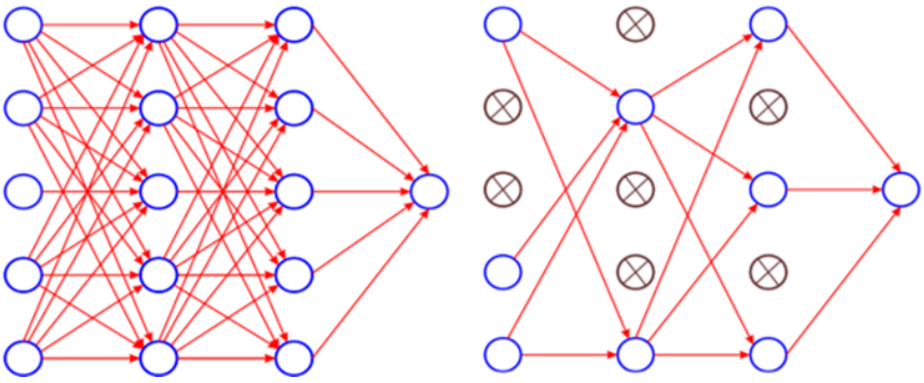
Fitur *map* yang dihasilkan pada tahap sebelumnya terbentuk *multi-dimensional array*. Sehingga, sebelum masuk pada tahap *Fully Connected Layer*, *Feature Map* tersebut akan melalui proses "*flatten*" atau *reshape*. Proses ini menghasilkan sebuah vektor yang digunakan sebagai *input* dari *Fully Connected Layer*. Proses *flatten* menghasilkan sebuah vektor yang akan digunakan sebagai *input* dari *Fully Connected Layer*. *Fully Connected Layer* terdiri dari beberapa *Hidden Layer*, *Action Function*, *Output Layer* dan *Loss Function* [4]. Pada gambar 2.6 menjelaskan tentang bagan *Fully Connected Layer*.



Gambar 2.6. *Fully Connected Layer*

2.3.4 *Dropout*

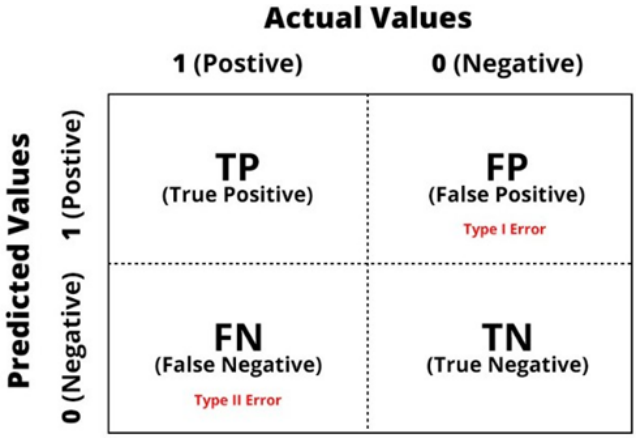
Dropout merupakan salah satu usaha untuk mencegah terjadinya *overfitting* dan juga mempercepat proses *learning* [14]. *Overfitting* merupakan kondisi dimana hampir semua data yang telah melalui proses *training* mencapai presentase yang baik, tetapi sering kali terjadi ketidaksesuaian dalam proses prediksinya. Dalam sistem kerjanya, *Dropout* menghilangkan sementara suatu neuron yang berupa *Hidden* Layer maupun *Visible Layer* yang berada didalam jaringan. Pada gambar 2.7 menjelaskan tentang bagan *Dropout*.



Gambar 2.7. Jaringan saraf sebelum dan sesudah *Dropout*

2.4 *Confusion Matrix*

Confusion Matrix adalah sebuah tabel yang dapat di-*generate* untuk *classifier* pada dataset dan dapat digunakan untuk menggambarkan performa *classifier* [15]. Pada gambar 2.8 menjelaskan tentang bagan *Confusion Matrix*.



Gambar 2.8. *Confusion Matrix*

Pada dasarnya, *Confusion Matrix* memberikan perbandingan antara hasil klasifikasi oleh model dengan hasil klasifikasi aktual untuk memberikan informasi yang lebih efektif. Berikut ini adalah gambar *confusion matrix* dengan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda :

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*, yaitu sebagai berikut:

1. *True Positive* (TP) yaitu, kondisi dimana data positif yang telah diprediksi dengan hasil benar.
2. *False Positive* (FP) yaitu, kondisi dimana data negatif diprediksi sebagai data positif.
3. *True Negative* (TN) yaitu, kondisi dimana data negatif yang diprediksi benar.
4. *False Negative* (FN) yaitu, kondisi dimana data positif tetapi diprediksi sebagai data negatif.

Berdasarkan istilah-istilah tersebut maka didapatkanlah rumusan *confusion matrix* untuk melakukan perhitungan mendapatkan akurasi, presisi, dan *recall* sebagai berikut :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.4}$$

$$Precision = \frac{TP}{TP+FP} \tag{2.5}$$

$$Recall = \frac{TP}{TP+FN} \tag{2.6}$$

$$F1\text{ - }Score = \frac{Recall*Precision}{Recall+Precision} \tag{2.7}$$

2.5 Dataset Wajah Yang Digunakan

2.5.1 Dataset *AT&T*

Data wajah *AT&T* berisi citra wajah yang diambil pada bulan April 1992 dan 1994. Terdapat sepuluh variasi wajah yang berbeda untuk masing-masing dari empat puluh subjek yang berbeda. Semua gambar wajah diambil dengan latar belakang yang gelap dan subjek berada dalam posisi tegak. dataset ini menggunakan format PGM dan setiap gambar memiliki ukuran 92×112 piksel, dengan setiap piksel memiliki 256 tingkat keabuan [16]. Pada gambar 2.9 merupakan contoh citra wajah dari dataset *AT&T*.

Gambar 2.9. Dataset *AT&T*

2.5.2 Dataset *Yale Face*

Data wajah *Yale Face* berisi 165 gambar *grayscale* dari 15 orang, dengan 11 gambar dari setiap orang dalam kondisi cahaya yang berbeda. Data gambar yang digunakan dalam penelitian ini diperoleh pada akhir tahun 1990-an dan memiliki resolusi yang relatif rendah, yaitu sekitar 320×243 piksel. Proses pengumpulan gambar dilakukan dalam lingkungan yang terkendali dengan kondisi standar, di mana latar belakang dan pose wajah tetap konsisten. Keberadaan kondisi tersebut membuktikan kegunaan dataset ini dalam pengujian algoritma yang dirancang untuk mengatasi variasi pencahayaan, pose, dan ekspresi wajah [17]. Pada gambar 2.10 merupakan contoh citra wajah dari dataset *Yale Face*.

Gambar 2.10. Dataset *Yale Face*

2.5.3 Dataset *Georgia Tech Face*

Data wajah *Georgia Tech* berisi gambar 50 orang yang diambil dalam dua atau tiga sesi antara 01 Juni 1999 dan 15 November 1999 di Pusat Pemrosesan Sinyal dan Gambar di Institut Teknologi Georgia [18]. Pada gambar 2.11 merupakan contoh citra wajah dari dataset *Georgia Tech*.

Gambar 2.11. Dataset *Georgia Tech*

Semua orang dalam *database* diwakili oleh 15 gambar JPEG berwarna dengan *background* berantakan yang diambil pada resolusi 640×480 piksel. Ukuran rata – rata wajah dalam gambar ini adalah 160×220 piksel. Gambar menunjukkan wajah depan dan miring dengan ekspresi wajah, kondisi pencahayaan, dan skala yang berbeda. Setiap gambar diberi label secara manual untuk menentukan posisi wajah dalam gambar.

2.5.4 Dataset Mandiri

Dataset Mandiri merupakan data citra wajah yang diambil secara mandiri. Terdapat 10 data citra wajah yang berbeda dari 70 subjek berbeda. Semua citra wajah diambil dengan posisi tegak dan berbeda ekspresi. dataset tersebut berformat JPG dengan ukuran setiap citra wajah adalah 92×112 piksel. Pada gambar 2.12 merupakan contoh citra wajah dari dataset Mandiri

Gambar 2.12. Dataset Mandiri

2.6 Python

Python adalah bahasa pemrograman tingkat tinggi yang digunakan dalam membuat aplikasi, perintah komputer dan menganalisis data. Python merupakan bahasa pemrograman yang tidak ada batasan dalam penyalinannya atau mendistribusikannya (*freeware*). Python digunakan untuk membuat program dan menyelesaikan permasalahan-permasalahan sebagai *general purpose language*. Python termasuk bahasa pemrograman tingkat tinggi yang mudah untuk dipelajari [19].