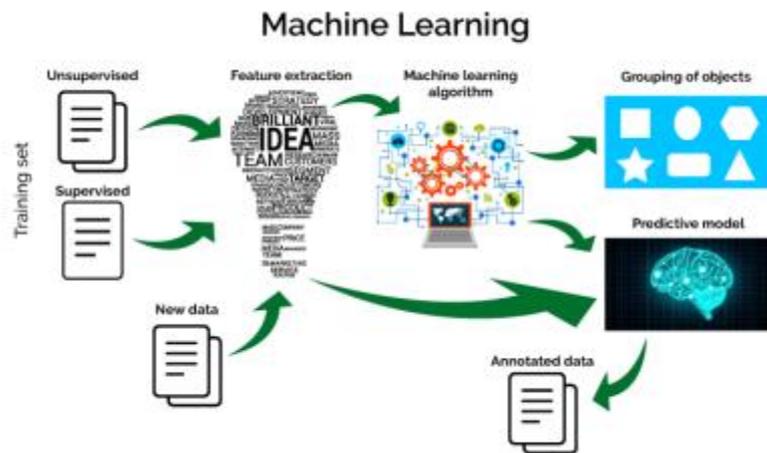


BAB II

LANDASAN TEORI

2.1 *Machine Learning*

Machine learning merupakan serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran (Danukusumo, 2017).



Gambar 2.1 *Machine Learning* (Pantech, 2018)

Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel pada tahun 1959. Menurut Arthur Samuel, *machine learning* adalah suatu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrograman yang jelas.

Menurut (Mohri et.al, 2012) *machine learning* dapat didefinisikan sebagai metode komputasi berdasarkan pengalaman untuk meningkatkan performa atau membuat prediksi yang akurat. Definisi pengalaman disini ialah informasi sebelumnya yang telah tersedia dan bisa dijadikan data pembelajar.

Dalam pembelajaran *machine learning*, terdapat skenario-skenario seperti :

1. *Supervised Learning*

Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu membuat prediksi dari data yang telah diberi label.

2. *Unsupervised Learning*

Penggunaan skenario *unsupervised learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu mencoba untuk mengelompokan data berdasarkan karakteristik-karakteristik yang ditemui.

3. *Reinforcement learning*

Pada skenario *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar.

2.2 *Deep Learning*

Deep Learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *deep learning* memberikan arsitektur yang sangat kuat untuk *supervised learning*. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada *machine learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra

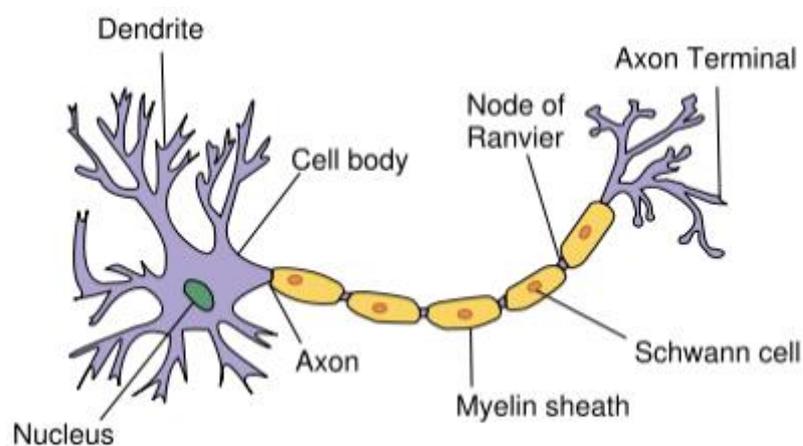
maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi.

Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditangguhkan pada algoritma *machine learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini terus berkembang hingga *deep learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer vision*, *Speech recognition*, dan *Natural Language Processing*. *Feature Engineering* adalah salah satu fitur utama dari *deep learning* untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature Engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga.

Dalam *deep learning*, metode CNN atau *Convolutional Neural Network* sangatlah bagus dalam menemukan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis nonlinier yang dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia *deep learning* penggunaan GPU sudah sangatlah umum (Danukusumo, 2017).

2.3 Artificial Neural Network

Artificial Neural Network (ANN) merupakan suatu model komputasi paralel yang meniru fungsi dari sistem jaringan syaraf biologi otak manusia. Dalam otak manusia terdiri dari milyaran *neuron* yang saling berhubungan. Hubungan ini disebut dengan *Synapses*. Komponen *neuron* terdiri dari satu inti sel yang akan melakukan pemrosesan informasi, satu akson (*axon*) dan minimal satu *dendrit*. Informasi yang masuk akan diterima oleh *dendrit*. Selain itu, *dendrit* juga menyertasi akson sebagai keluaran dari suatu pemrosesan informasi.



Gambar 2.2 Jaringan Syaraf Manusia

Cara kerja dari sistem syaraf pada gambar 2.2 adalah bermula pada sinyal masuk melalui *dendrit* menuju *cell body*. Kemudian sinyal akan di proses didalam *cell body* berdasarkan fungsi tertentu (*Summation Proses*). Jika sinyal hasil proses melebihi nilai ambang batas (*threshold*) tertentu maka sinyal tersebut akan membangkitkan *neuron* untuk meneruskan sinyal tersebut. Sedangkan jika dibawah nilai ambang batasnya maka sinyal tersebut akan dihalangi (*inhibited*).

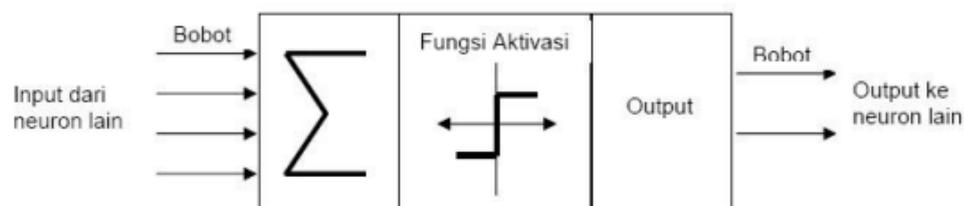
Kemudian sinyal yang diteruskan akan menuju ke *axon* dan akhirnya menuju ke *neuron* lainnya melewati *synapses*.

ANN merupakan sistem adatif yang dapat mengubah strukturnya untuk memecahkan suatu masalah berdasarkan informasi internal maupun eksternal. ANN bersifat fleksibel terhadap inputan data dan menghasilkan *output* respon konsisten (Pham, 1994). ANN telah banyak digunakan dalam area yang luas. (Kumar, 2003) menjelaskan, penerapan ANN dapat mengidentifikasi beberapa aplikasi yaitu:

1. Estimasi/prediksi (aproksimasi fungsi, peramalah)
2. Pengenalan Pola (klasifikasi, diagnosis, dan analisis diskriminan)
3. Klustering (pengelompokan tanpa adanya pengetahuan sebelumnya).

2.4 Struktur *Neural Network*

Dari struktur *neuron* pada otak manusia, dan proses kerja yang dijelaskan pada gambar 2.2, maka konsep dasar pembangunan *neural network* buatan (*Artificial Neural Network*) terbentuk. Ide mendasar dari *Artificial Neural Network* (ANN) adalah mengadopsi mekanisme berpikir sebuah sistem atau aplikasi yang menyerupai otak manusia, baik untuk pemrosesan berbagai sinyal elemen yang diterima, toleransi terhadap kesalahan/*error*, dan juga *parallel processing*.



Gambar 2.3 Struktur *Neural Network*

Gambar 2.3 menunjukkan struktur yang dimiliki oleh *Neural Network*.

Komponen yang dimiliki struktur tersebut sebagai berikut :

1. Input terdiri dari variabel independet ($X_1, X_2, X_3, \dots, X_n$) yang merupakan sebuah sinyal yang masuk ke sel syaraf.
2. Bobot (*Weight*) terdiri dari beberapa bobot ($W_1, W_2, W_3, \dots, W_n$) yang berhubungan dengan masing-masing *node*.
3. *Threshold* merupakan nilai ambang batas internal dari *node*. Besar nilai ini mempengaruhi aktivasi dari *output node* y .
4. *Activation Function* (Fungsi Aktivasi) merupakan operasi matematika yang dikenal pada sinyal *output* y .

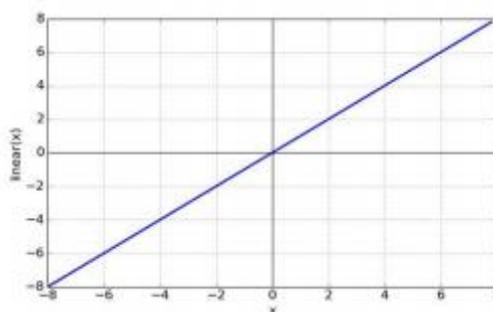
Cara kerja struktur *neural network* diatas tidak jauh berbeda dengan struktur jaringan syaraf pada manusia. Informasi (*input*) akan dikirimkan dengan bobot kedatangan tertentu. *Input* tersebut kemudian diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap *neuron*. Jika input tersebut melewati suatu nilai ambang tertentu, maka *neuron* tersebut akan diaktifkan. Jika tidak, *neuron* tersebut tidak akan diaktifkan. Apabila *neuron* diaktifkan, selanjutnya *neuron* tersebut akan mengirimkan *output* melalui bobot-bobot *outputnya* ke semua *neuron* yang berhubungan dengannya, begitu seterusnya. Pada *neuron layer*, penempatan *neuron-neuron* akan dikumpulkan dalam *neuron layer* (lapisan-lapisan). Kemudian *neuron-neuron* pada satu lapisan akan di hubungkan dengan lapisan-lapisan sebelum dan sesudahnya, kecuali lapisan *input*

dan *output*. Informasi yang di bawa dari langkah *input* awal akan dirambatkan dari lapisan ke lapisan dari lapisan *input* sampai lapisan *output*. Lapisan ini sering disebut dengan istilah *hidden layer* (lapisan tersembunyi). Pada umumnya setiap *neuron* terletak pada lapisan yang sama akan memiliki keadaan yang sama. Sehingga pada setiap lapisan sama, setiap *neuron* akan memiliki fungsi aktivasi yang sama. Koneksi antar lapisan dengan *neuron* harus selalu berhubungan. Faktor terpenting dalam menentukan kelakuan suatu *neuron* adalah terletak pada pola bobot dan fungsi aktivasinya.

2.5 Activation Function

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*.

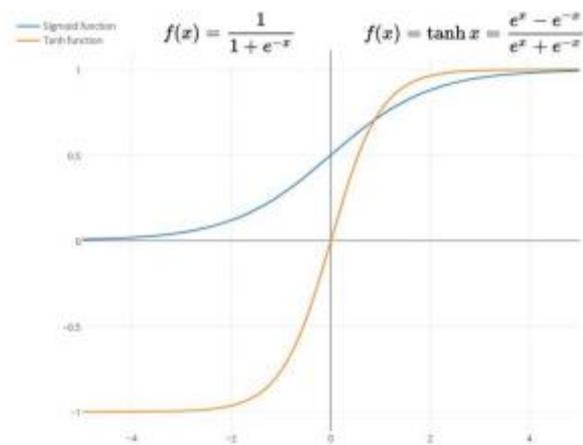
2.5.1. Linier Function



Gambar 2.4 *Linier Function*

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*.

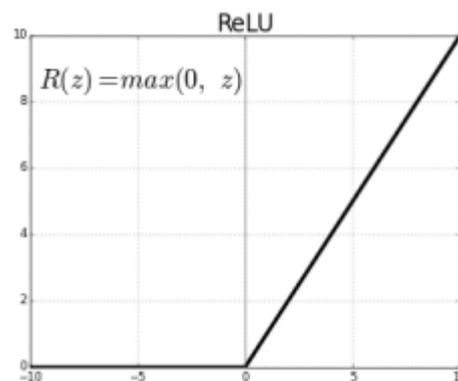
2.5.2. Sigmoid dan Tanh Function (Non Linier)



Gambar 2.5 Sigmoid dan 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.

2.5.3. ReLU (Non Linier)

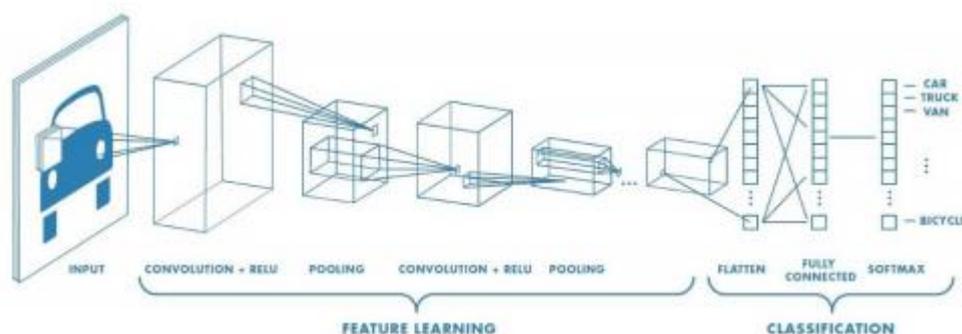


Gambar 2.6 ReLU

Pada dasarnya *ReLU* melakukan “*threshold*” dari 0 hingga *infinity*. *ReLU* juga dapat menutupi kelemahan yang dimiliki oleh *Sigmoid* dan *Tanh*.

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. CNN ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada dasarnya klasifikasi citra dapat digunakan dengan MLP, akan tetapi dengan metode MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. Penelitian awal yang mendasari penemuan CNN ini pertama kali dilakukan oleh Hubel dan Wiesel (Hubel & Wiesel, T, 1968) mengenai *virtual cortex* pada indera penglihatan kucing. Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi layer dan pooling layer. Berikut adalah jaringan arsitektur *Convolutional Neural Network* :



Gambar 2.7 Arsitektur *Convolutional Neural Network*

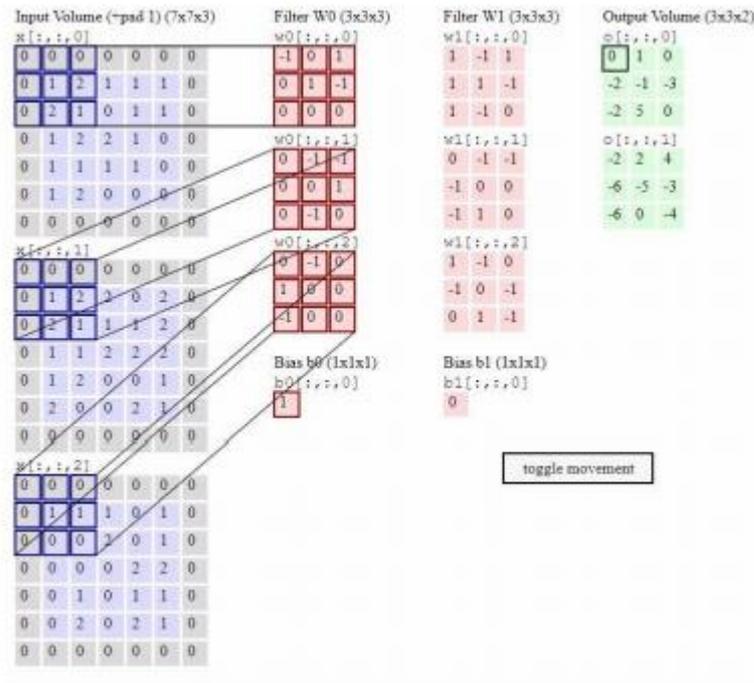
Berdasarkan gambar 2.7, Tahap pertama pada arsitektur CNN adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi *ReLU* (*Rectifier Linear Unit*), Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses pooling. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

2.6.1. Convolution Layer

Convolution layer merupakan bagian dari tahap pada arsitektur CNN. Tahap ini melakukan operasi konvolusi pada *output* dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari jaringan arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *feature map* dari *input* citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil.

Convolutional Layer terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Sebagai contoh, layer pertama pada *feature extraction layer* biasanya adalah conv. Layer dengan ukuran 5x5x3. Panjang 5 pixels, tinggi 5 pixels dan tebal/jumlah 3 buah sesuai dengan channel dari image tersebut. Ketiga filter ini akan digeser keseluruh

bagian dari gambar. Setiap pergeseran akan dilakukan operasi “dot” antara *input* dan nilai dari filter tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*. Perhatikan ilustrasi berikut :

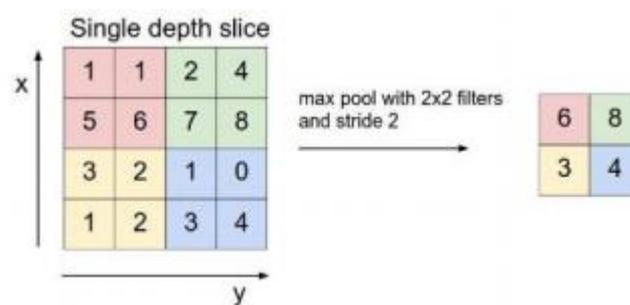


Gambar 2.8 *Convolutional layer*
(Medium Samuel Senna, 2017)

2.6.2. Operasi *Pooling*

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling Layer* biasanya berada setelah conv. Pada dasarnya *pooling* layer terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max-pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *Pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur

model CNN dapat secara progresif mengurangi ukuran volume *output* pada *feature map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *overfitting*. Lapisan *pooling* bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari *inputnya*. Berikut ini adalah contoh gambar operasi *max-pooling* :

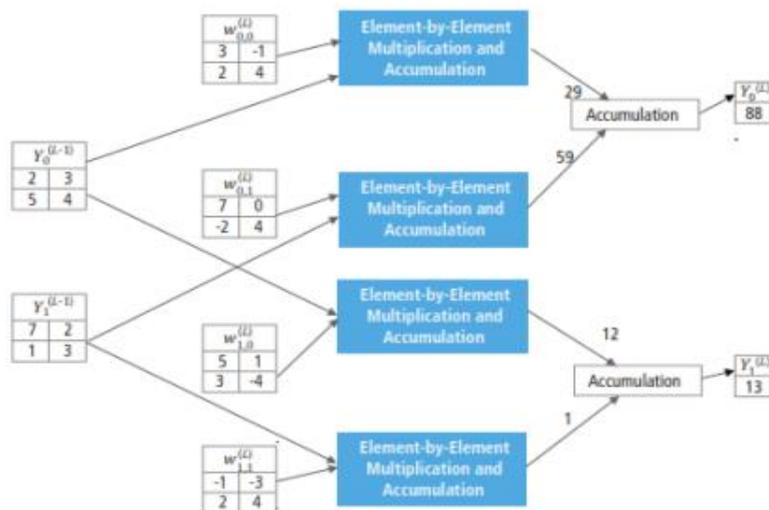


Gambar 2.9 Operasi *max-pooling*
(Medium Samuel Senna, 2017)

Gambar 2.9 menunjukkan proses dari *max-pooling*. *Output* dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan *pooling* diatas akan beroperasi pada setiap irisan kedalaman volume *input* secara bergantian. Jika dilihat dari gambar diatas operasi *max-pooling* dengan menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4, dari masing-masing 4 angka pada input operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2.

2.6.3. Fully-Connected Layer

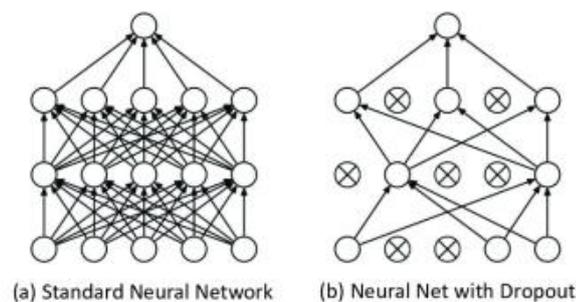
Fully-Connected Layer adalah sebuah lapisan dimana semua *neuron* aktivasi dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan selanjutnya sama seperti halnya dengan *neural network* biasa. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara *linier*. Perbedaan antara lapisan *fully-connected* dan lapisan konvolusi biasa adalah *neuron* di lapisan konvolusi terhubung hanya ke daerah tertentu pada *input*, sementara lapisan *fully-connected* memiliki *neuron* yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda. Berikut ini adalah proses *fully-connected* :



Gambar 2.10 Processing of Fully-Connected Layer

2.6.4. Dropout Regulation

Dropout merupakan sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa *neuron* secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain *neuron-neuron* tersebut dibuang secara acak. Hal ini berarti bahwa kontribusi *neuron* yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada *neuron* pada saat melakukan *backpropagation*. Berikut adalah gambar proses *dropout*:



Gambar 2.11 *Dropout Regulation*

Berdasarkan gambar 2.11, pada bagian a merupakan jaringan syaraf biasa yang memiliki dua *hidden layer*. Sedangkan pada bagian b merupakan jaringan syaraf dengan menggunakan *dropout*. Dari gambar tersebut terlihat terdapat beberapa neuron aktivasi yang tidak dipakai lagi.

2.7 Softmax Classifier

Softmax Classifier adalah generalisasi dari fungsi logistik. *Output* dari *softmax* ini dapat digunakan untuk mewakili distribusi sebuah kategori. *Softmax function* digunakan dalam berbagai macam metode klasifikasi contohnya *multinomial logistic regression*, *multiclass linear discriminant analysis*, *naive Bayes classifier*, dan *neural network*. Secara spesifiknya fungsi ini biasa

digunakan pada metode klasifikasi *multinomial logistic regression* dan *multiclass linear discriminant analysis*. Berikut adalah fungsi yang diberikan :

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

Gambar 2.12 Fungsi *Softmax Classifier*

Notasi f_j menunjukkan hasil fungsi untuk setiap elemen ke- j pada vektor keluaran kelas. Argumen z adalah hipotesis yang diberikan oleh model pelatihan agar dapat diklasifikasi oleh fungsi *softmax*. *Softmax* juga memberikan hasil yang lebih intuitif dan juga memiliki interpretasi probabilistik yang lebih baik dibanding algoritma klasifikasi lainnya. *Softmax* memungkinkan kita untuk menghitung probabilitas untuk semua label. Dari label yang ada akan diambil sebuah vektor nilai bernilai riil dan merubahnya menjadi vektor dengan nilai antara nol dan satu yang bila semua dijumlah akan bernilai satu.

2.8 Data Augmentation

Data Augmentation adalah teknik yang memperkenalkan sampel baru klasifikasi dengan menerapkan transformasi pada sampel nyata. Penelitian (Shijie & Ping n.d.) menyatakan *Data Augmentation method* dibagi menjadi dua yaitu :

2.8.1. Unsupervised Data Augmentation

Unsupervised Data Augmentation adalah metode augmentasi yang tidak terkait dengan data label. Yang termasuk kedalam *unsupervised data augmentation* adalah sebagai berikut :

1) *Cropping*

Cropping adalah metode augmentasi yang dilakukan dengan cara memangkas bagian dari gambar asli dan merubah ukuran gambar yang dipangkas ke resolusi tertentu.

2) *Flipping*

Flipping adalah metode augmentasi yang dilakukan dengan cara membalikan gambar secara horizontal maupun vertikal.

3) *Rotation*

Rotation adalah metode augmentasi yang dilakukan dengan cara memutar gambar secara acak.

4) *Shifting*

Shifting adalah metode augmentasi yang dilakukan dengan cara gambar digeser kearah kiri atau kanan, jarak terjemahan dan panjang langkah dapat ditentukan secara manual untuk mengubah lokasi konten gambar.

5) *Color Jittering*

Color Jittering adalah metode augmentasi yang dilakukan dengan cara mengubah secara acak faktor saturasi warna, kecerahan dan kontras dalam ruang warna gambar.

6) *Noise*

Noise adalah metode augmentasi yang dilakukan dengan cara menambahkan *noise* secara acak ke saluran RGB dari setiap pixel dalam gambar.

7) *PCA Jittering*

PCA (Principal Component analysis) Jittering adalah metode augmentasi yang dilakukan dengan cara melakukan PCA pada gambar untuk mendapatkan komponen utama, kemudian ditambahkan ke gambar asli dengan gangguan *gaussian* sebesar (0,0,1) untuk menghasilkan gambar baru.

2.8.2. Supervised Data Augmentation

Supervised Data Augmentation adalah metode augmentasi yang terkait dengan data label. Yang termasuk kedalam metode supervised data augmentation adalah :

1) *Generative Adversarial Netowrk (GAN)*.

GAN model terdiri dari model Generatif (G) dan model Diskriminatif (D). Dalam proses pelatihan, G diajarkan untuk memetakan dari ruang laten ke distribusi data tertentu yang menarik dan D secara bersamaan diajarkan untuk membedakan antara contoh dari data yang benar.

2.9 Literature Review

Tabel 2.1 Literature Review

No	Peneliti / Tahun	Judul	Problem	Metode / Algoritma / Teknik / Model / Sensor / Platform	State Of The Art / Keterbaruan
1.	(Wang & Perez, 2017)	<i>The Effectiveness of Data Augmentation in Image Classification using Deep Learning</i>	Bagaimana mengambil data terstruktur yang kecil dan diperbesar dengan cara yang meningkatkan kinerja model yang dilatih di dalamnya.	Metode : GAN, <i>basic transformations</i> Model : CycleGAN Algoritma : CNN-5 layer	Menggunakan metode <i>neural augmentation</i> .
2.	(Shi, Wang, Ding, Yang, & Li, 2018)	<i>Data Augmentation with Improved Generative Adversarial Networks</i>	Tidak adanya label dan kualitas data yang pasti.	Metode : <i>Generative Adversarial Network</i>	<i>Improve metode Data Augmentation GAN</i>
3.	(Takahashi, Matsubara, & Uehara, 2015)	<i>Data Augmentation using Random Image Cropping and Patching for Deep CNNs</i>	<i>Convolutional Neural Network (CNN)</i> mencapai hasil luar biasa dalam tugas pemrosesan gambar berisiko <i>overfitting</i> .	Metode : <i>Random Cropping, Random Patching, dropout, cutout, random erasing.</i> Algoritma : <i>Convolutional Neural Network (CNN)</i>	Memperkenalkan metode <i>Data Augmentation</i> baru yaitu RICAP (<i>Random Cropping and Patching</i>)
4.	Mikołajczyk & Grochowski,	<i>Data augmentation for</i>	Keseimbangan kelas yang tidak merata	Metode : <i>Affine transformations, GAN,</i>	mempbandingkan dan menganalisis beberapa

	2018)	<i>improving deep learning in image classification problem</i>	dalam dataset.	<i>texture transfer</i> Algoritma : CNN Model : VGG 16 architecture	metode augmentasi data dalam tugas klasifikasi gambar
5.	(Shijie & Ping, n.d.)	<i>Research on Data Augmentation for Image Classification Based on Convolution Neural Networks</i>	Kinerja jaringan saraf konvolusi yang dalam akan lebih ditingkatkan dengan perluasan set data pelatihan	Metode : GAN, <i>Cropping, Shifting, flipping, PCA jittering, noise, rotation, beberapa kombinasi, color jittering</i> Algoritma : <i>Convolutional Neural Network (CNN)</i>	mengeksplorasi dampak dari berbagai metode augmentasi data pada tugas-tugas klasifikasi gambar dengan jaringan Neural konvolusi yang mendalam
6.	(Sun, Shrivastava, Singh, & Gupta, 2011)	<i>Revisiting Unreasonable Effectiveness of Data in Deep Learning Era</i>	Apa yang akan terjadi jika meningkatkan ukuran dataset sebesar $10 \times$ atau $100 \times$?	Model : ImegNet, Pascal VOC, JFT-300M, ResNet-50, ResNet-152, COCO.	menyajikan hasil canggih di beberapa tolok ukur menggunakan model yang dipelajari dari JFT-300M.
7.	(Alhussein Fawzi, Horst Samulowitz, n.d.)	<i>Adaptive Data Augmentation For Image Classification</i>	Mencari transformasi kecil yang menghasilkan kerugian klasifikasi maksimal pada sampel yang diubah	Metode : <i>Affine Data Augmentation, Adaptive Data Augmentation</i> Algoritma : <i>Convolutional Neural Network</i> Dataset : MNIST, Small NORB	Hasil eksperimen pada dua set data menunjukkan bahwa skema sederhana ini menghasilkan hasil yang setara (atau superior) dengan metode canggih.
8.	(Zhong, Zheng, Kang, Li, & Yang, n.d.)	<i>Random Erasing Data Augmentation</i>	Model CNN gagal secara drastis saat memprediksi data baru	Metode : <i>Random Erasing, Random Cropping, Dropout, Random Noise</i> Algoritma : CNN	memperkenalkan Random Erasing, metode augmentasi data baru untuk melatih <i>Convolutional Neural Network (CNN)</i> .

				Model : Pascal VOC 2007 Pendekatan : <i>Detailed Procedure of Random Erasing</i>	
9.	(P, Wijaya, & Soelaiman, 2016)	Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101	Bagaimana menduplikasi kemampuan manusia dalam memahami informasi citra, agar komputer dapat mengenali objek pada citra selayaknya manusia.	Algoritma : <i>Convolutional Neural Network</i>	klasifikasi citra objek dengan tingkat <i>confusion</i> yang berbeda pada basis data Caltech 101 menghasilkan rata-rata nilai akurasi mencapai.
10.	(Zeiler & Fergus, 2014)	<i>Visualizing and Understanding Convolutional Networks</i>	Tidak ada pemahaman yang jelas tentang mengapa kinerja CNN sangat baik, atau bagaimana CNN dapat ditingkatkan.	Model : ImegNet, Convnet, Deconvnet, Pascal VOC Algoritma : CNN	Memperkenalkan teknik visualisasi baru yang memberikan wawasan tentang fungsi lapisan fitur menengah dan pengoperasian classifier
11.	(Guo, Dong, & Li, 2017)	<i>Simple Convolutional Neural Network on Image Classification</i>	Bagaimana mengetahui model terbaik yang biasa digunakan dalam pembelajaran mendalam	Metode : <i>DropConnect, multi column DNN, Augmented Pattern Classification, max-pooling, RCNN,</i> Algoritma : CNN	mengusulkan jaringan saraf <i>Convolutional</i> sederhana pada klasifikasi gambar.
12.	Yanagisawa, Yamashita, &	<i>A Study on Object Detection Method</i>	Tidak jelasnya apakah metode berfungsi	Algoritma : CNN, Fast-RCNN, SSD	menguji efektivitas deteksi objek manga dengan

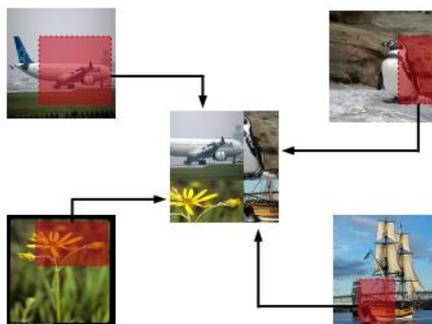
	Watanabe, n.d.)	<i>from Manga Images using CNN</i>	dengan baik untuk gambar manga atau tidak karena fitur gambar tersebut berbeda dari gambar alami		membandingkan Fast R-CNN, R-CNN, dan SSD yang lebih cepat.
13.	(Santoso et al. n.d.)	IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK PENGENALAN WAJAH	Adanya permasalahan dalam faktor pencahayaan, ekspresi wajah dan perubahan atribut pada wajah	Algoritma : CNN	Pembuatan aplikasi pengenalan wajah dengan algoritma CNN menggunakan librari keras.
14.	(Feng, Li, Yang, & Yan, n.d.)	<i>V-CNN: Data Visualizing based Convolutional Neural Network</i>	Efisiensi data pelatihan pada CNN terlalu rendah	Algoritma : CNN, Visualisasi CNN (V-CNN)	"Membuat model data fit," mengedepankan CNN berdasarkan visualisasi data, bernama V-CNN
15.	(Wicaksono, Suciati, Faticah, Uchimura, & Koutaki, 2017)	<i>Modified Convolutional Neural Network Architecture for Batik Motif Image Classification</i>	Tidak mengetahui motif batik yang dikenakan dan informasi batik yang terkandung dalam gambar batik	Algoritma : CNN metode : CNN GoogleNet, CNN InRes	menawarkan sistem klasifikasi gambar motif batik menggunakan metode CNN dengan arsitektur jaringan baru yang dikembangkan dengan menggabungkan GoogLeNet dan Residual Networks bernama IncRes.

2.10 Penelitian Terdekat

Tabel 2.2 Penelitian Terdekat

No	Peneliti / Tahun	Judul	Problem	Metode / Algoritma / Teknik / Model / Sensor / Platform	State Of The Art / Keterbaruan
1.	(Takahashi, Matsubara, & Uehara, 2015)	<i>Data Augmentation using Random Image Cropping and Patching for Deep CNNs</i>	<i>Convolutional Neural Network (CNN)</i> mencapai hasil luar biasa dalam tugas pemrosesan gambar berisiko <i>overfitting</i> .	Metode : <i>Random Cropping, Random Patching, dropout, cutout, random erasing.</i> Algoritma : <i>Convolutional Neural Network (CNN)</i>	Memperkenalkan metode <i>Data Augmentation</i> baru yaitu <i>RICAP (Random Cropping and Patching)</i>
2.	(Shijie & Ping, n.d.)	<i>Research on Data Augmentation for Image Classification Based on Convolution Neural Networks</i>	Kinerja jaringan saraf konvolusi yang dalam akan lebih ditingkatkan dengan perluasan set data pelatihan	Metode : <i>GAN, Cropping, Shifting, flipping, PCA jittering, noise, rotation, beberapa kombinasi, color jittering</i> Algoritma : <i>Convolutional Neural Network (CNN)</i>	mengeksplorasi dampak dari berbagai metode augmentasi data pada tugas-tugas klasifikasi gambar dengan jaringan Neural konvolusi yang mendalam, di mana Alexnet digunakan sebagai model jaringan pra-pelatihan dan subset dari CIFAR10 dan ImageNet (10 kategori) dipilih sebagai kumpulan data asli

Tabel 2.2 menjelaskan beberapa penelitian terkait yang dijadikan acuan terkait peningkatan akurasi algoritma *Convolutional Neural Network* (CNN) dengan metode *Data Augmentation Traditional Transformations*. Penelitian berjudul “*Data Augmentation using Random Image Cropping and Patching for Deep CNNs*” yang dilakukan (Takahashi, Matsubara, & Uehara, 2015) memperkenalkan metode baru *Data Augmentation* yaitu RICAP (*Random Cropping and Patching*). Metode ini merupakan gabungan antara *Data Augmentation Cropping* dengan *Patching*.



Gambar 2.13 *Random Cropping and patching*

Penelitian berjudul “*Research on Data Augmentation for Image Classification Based on Convolution Neural Networks*” yang dilakukan (Shijie & Ping, n.d.) mengeksplorasi dampak dari berbagai metode augmentasi data pada tugas-tugas klasifikasi gambar dengan jaringan *Neural Network*. Berdasarkan dua penelitian yang telah disebutkan, maka diusulkan penelitian yang berjudul “*Penggunaan Metode Traditional Transformations Data Augmentation Untuk Peningkatan Hasil Akurasi Pada Model Algoritma Convolutional Neural Network (CNN) Di Klasifikasi Gambar*”.

2.11 Matriks Penelitian

Tabel 2.3 Matrik Penelitian

No	Peneliti / Tahun	Judul	Ruang Lingkup Penelitian					
			Pendekatan	Metode	Algoritma	Platform	Model	Modul
1.	(Wang & Perez, 2017)	<i>The Effectiveness of Data Augmentation in Image Classification using Deep Learning</i>		✓	✓		✓	
2.	(Shi, Wang, Ding, Yang, & Li, 2018)	<i>Data Augmentation with Improved Generative Adversarial Networks</i>		✓				
3.	(Takahashi, Matsubara, & Uehara, 2015)	<i>Data Augmentation using Random Image Cropping and Patching for Deep CNNs</i>		✓	✓			
4.	Mikołajczyk & Grochowski, 2018)	<i>Data augmentation for improving deep learning in image classification problem</i>		✓	✓		✓	
5.	(Shijie & Ping, n.d.)	<i>Research on Data Augmentation for Image Classification Based on Convolution Neural Networks</i>		✓	✓			
6.	(Sun, Shrivastava, Singh, & Gupta, 2011)	<i>Revisiting Unreasonable Effectiveness of Data in Deep Learning Era</i>					✓	
7.	(Alhussein Fawzi , Horst	<i>Adaptive Data Augmentation For Image Classification</i>		✓	✓			

	Samulowitz, n.d.)							
8.	(Zhong, Zheng, Kang, Li, & Yang, n.d.)	<i>Random Erasing Data Augmentation</i>	✓	✓	✓		✓	
9.	(P, Wijaya, & Soelaiman, 2016)	Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101			✓			
10.	(Zeiler & Fergus, 2014)	<i>Visualizing and Understanding Convolutional Networks</i>			✓		✓	
11.	(Guo, Dong, & Li, 2017)	<i>Simple Convolutional Neural Network on Image Classification</i>		✓	✓			
12.	Yanagisawa, Yamashita, & Watanabe, n.d.)	<i>A Study on Object Detection Method from Manga Images using CNN</i>			✓			
13.	(Santoso et al. n.d.)	IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK PENGENALAN WAJAH			✓			
14.	(Feng, Li, Yang, & Yan, n.d.)	<i>V-CNN: Data Visualizing based Convolutional Neural Network</i>			✓			
15.	(Wicaksono, Suciati, Faticah, Uchimura, & Koutaki, 2017)	<i>Modified Convolutional Neural Network Architecture for Batik Motif Image Classification</i>		✓	✓			
16.	(Deni Fermansah , 2019)	Penggunaan Metode <i>Traditional Transformations Data Augmentation</i> Untuk Peningkatan Hasil Akurasi Pada Model Algoritma <i>Convolutional Neural Network (CNN)</i> Di Klasifikasi Gambar.		✓	✓		✓	

Berdasarkan tabel matriks penelitian 2.3, penelitian ini membahas tentang metode , algoritma, dan model.