

BAB II

TUNJAUAN PUSTAKA DAN LANDASAN TEORI

1.1 Tinjauan Pustaka

Teknologi pengenalan gambar merupakan teknologi yang merujuk pada teknologi pemrosesan gambar secara digital dan menggunakan teknologi kecerdasan buatan, khususnya metode machine learning untuk membuat computer mengenali informasi yang berada pada gambar. Pengenalan gambar merupakan salah satu bidang utama dari pengenalan pola (Zhou et al., 2017). Pada penelitian ini pengenalan gambar diterapkan pada gambar wayang. Dimana wayang Indonesia memiliki bentuk dan ciri yang berbeda-beda sehingga dapat dikenali perbedaan dari masing-masing tokohnya (July and September, 2014). Permasalahan pengenalan pola tingkat keakuratan merupakan parameter penentu sukses atau tidaknya pengenalan tersebut. Keakuratan pengenalan pola dipengaruhi oleh beberapa faktor, salah satunya adalah jumlah data pelatihan yang sedikit dan model yang digunakan untuk pengenalan pola. Jumlah data pelatihan yang sedikit menyebabkan overfitting yang berpengaruh pada keakuratan pengenalannya. Salah satu cara untuk mengatasinya adalah dengan melakukan penambahan jumlah data melalui augmentasi data.

Beberapa penelitian yang telah dilakukan dalam pengenalan pola wayang telah dilakukan. (Grahita, Komma dan Kushiya, 2013) melakukan penelitian pada wayang pacitan. Penelitian mereka berfokus pada gaya visual dari wayang pacitan. Penulis melakukan penelitian yang sejenis yakni pengenalan pola wayang. Penulis menggunakan metode *deep learning*. Salah satu algoritma yang diterapkan adalah *Convolutional Neural Network*.

Deep learning memberikan model komputasi yang tersusun dari beberapa lapisan pemrosesan untuk mempelajari representasi data dengan berbagai tingkat abstraksi. Metode ini telah meningkatkan kemampuan dalam pengenalan pola, pengenalan suara, deteksi objek dan banyak domain lainnya (McCann, Jin dan Unser, 2017).

Convolutional Neural Network merupakan teknologi yang dikombinasikan dengan jaringan syaraf buatan dan *deep learning*. Algoritma ini telah diaplikasikan

pada banyak kasus pengenalan citra yang telah banyak menarik perhatian peneliti. *Convolutional neural network* terinspirasi dari struktur sistem visual yang dikemukakan oleh Hubel dan Wiesel (Deng, 2018). *Convolutional neural network* merupakan jenis dari jaringan syaraf buatan multilapisan yang didesain khusus untuk memproses data dua dimensi (McCann, Jin dan Unser, 2017). *Convolutional neural network* dapat mengurangi jumlah parameter pada jaringan untuk meningkatkan algoritma backpropagation sebagai jaringan propagasi maju. Pada algoritma *convolutional neural network*, area kecil yang juga disebut sebagai area *local sensing* diambil sebagai data input. Melalui propagasi maju, informasi melewati berbagai lapisan pada jaringan. Masing-masing lapisan terdiri dari *kernel* sehingga memperoleh beberapa fitur penting dari data yang diamati. Metode ini juga dapat memberikan bentuk bervariasi seperti *stretching* dan *rotation* (Salamon dan Bello, 2017). Hubungan dari tingkatan lapisan konvolusi pada convolutional neural network sangat baik digunakan untuk pemrosesan gambar dan dapat meng-ekstrak karakteristik dari gambar (Voulodimos, Doulamis, Doulamis dan Protopapadakis, 2018). Convolutional neural network telah diaplikasikan pada berbagai aspek, sebagai pengenalan wajah, analisis dokumen, deteksi suara dan pengenalan nomor plat kendaraan (Zarandy, Horvath dan Szolgay, 2018). CNN merupakan model yang memiliki kemampuan dalam melakukan pelatihan pada ukuran data yang besar. Elleuch et al. menunjukkan dengan menggunakan ReLU dan dropout meningkatkan performa dari *deep neural network* Dropout dapat mengurangi overfitting dengan secara acak mengurangi unit koneksi saat pelatihan (Elleuch et al., 2016), sehingga mencegah korelasi antara unit. Pada tahun 2015 Ioffe et al. memperkenalkan *batch normalization* yang merupakan teknik lain untuk membuat jaringan lebih kuat (Liu et al., 2018). Untuk menghindari masalah bahwa neuron menerima masukan dari berbagai distribusi pada setiap laluan dan harus menyesuaikan bobotnya. Augmentasi data merupakan *model-agnostic* untuk metode pembelajaran *invariance*. *Flipping* dan pemotongan secara acak telah ditunjukkan pada augmentasi data untuk meningkatkan kekuatan jaringan. Metode lain yang digunakan untuk meningkatkan kekuatan jaringan CNN adalah menggunakan lapisan *recurrent* yang disusun oleh Liang et al (Ming Liang dan

Xiaolin Hu, 2015). Ini akan membuat jaringan lebih dalam dan membuat akurasi semakin tinggi dengan nilai parameter yang sama.

1.2 Landasan Teori

1.2.1 Kesenian Wayang

Wayang adalah seni pertunjukan berupa drama yang khas. Seni pertunjukan ini meliputi seni suara, seni sastra, seni musik, seni tutur, seni rupa, dan lain-lain. Ada pihak beranggapan, bahwa pertunjukan wayang bukan sekedar kesenian, tetapi mengandung lambang-lambang keramat. Sejak abad ke-19 sampai dengan sekarang, wayang telah menjadi pokok bahasan serta dideskripsikan oleh para ahli. Wayang adalah sebuah kata bahasa Indonesia (Jawa) asli berarti bayang atau bayang-bayang, berasal dari akar yang dengan mendapat awalan wa menjadi kata wayang.

Kata-kata di dalam bahasa Jawa yang mempunyai akar kata yang dengan berbagai variasi vokal, antara lain: layang, dhoyong, puyeng, reyong, yang berarti selalu bergerak, tidak tetap, samar-samar, dan sayup-sayup. Kata wayang dan hamayang dahulu berarti mempertunjukkan bayangan, kemudian berkembang menjadi pertunjukan bayang-bayang dan menjadi seni pentas bayang-bayang atau wayang.

Wayang adalah sebuah wiracarita yang pada intinya mengisahkan kepahlawanan para tokoh yang berwatak baik menghadapi dan menumpas tokoh yang berwatak jahat. Kenyataan bahwa wayang yang telah melewati berbagai peristiwa sejarah, dari generasi ke generasi, menunjukkan betapa budaya pewayangan telah melekat dan menjadi bagian hidup bangsa Indonesia khususnya Jawa. Usia yang demikian panjang dan kenyataan bahwa hingga dewasa ini masih banyak orang yang menggemarnya menunjukkan betapa tinggi nilai dan berartinya wayang bagi kehidupan masyarakat. Wayang merupakan sastra tradisional yang memenuhi kualifikasi karya master piece, karya sastra dan atau budaya adiluhung.

1.2.2 Pengenalan Pola

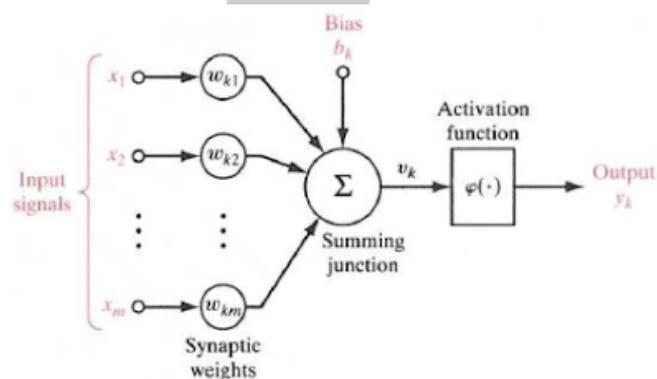
Pengenalan pola merupakan salah satu masalah klasifikasi dari visi komputer dan sering dikatakan sebagai tugas yang sulit. Yang menyebabkan pengenalan pola menjadi suatu permasalahan yang berbeda adalah melibatkan area pada gambar dan mengklasifikasikannya. Untuk mengenali sebuah gambar, maka perlu diketahui pula bagaimana gambar tersebut disegmentasikan. Beberapa fitur visual yang berbeda, seperti baju dan wajah manusia mungkin bagian objek yang sama namun sangat sulit untuk mengetahui perbedaannya tanpa mengenali objeknya terlebih dahulu. Pengenalan pola merupakan suatu kasus yang menjadikan suatu sistem dapat mengenali suatu objek yang didasarkan atas pengetahuan yang didapatkan sebelumnya oleh sistem tersebut. Sistem akan mencari kemiripan atas pengetahuan yang dimiliki terhadap pola baru yang akan dia kenali. Sehingga untuk mengetahui suatu pola dengan akurat, maka pengetahuan suatu sistem harus ditambahkan dengan menambah banyak gambar untuk pelatihan. Dengan demikian sistem dapat mengenali objek baru dengan baik dan benar.

1.2.3 Anatomi Dari Jaringan Syaraf Tiruan

Berikut anatomi dari jaringan syaraf tiruan (Jian, 1996).

- Lapisan*, yang digabungkan menjadi jaringan (model)
- Data masukan dan target yang sesuai
- Loss Function*, yang mendefinisikan sinyal umpan balik yang digunakan untuk proses belajar.
- Optimizer* yang menentukan bagaimana hasil belajar.

Gambar 2.1 menunjukkan jaringan syaraf tiruan.



Gambar 2. 1. Anatomi Jaringan Syaraf Tiruan

1.2.4 Normalisasi Gambar

Sebelum data diolah oleh CNN, maka penulis melakukan proses normalisasi data. Dalam proses ini dilakukan penyamaan ukuran gambar pada dataset. Pada penelitian ini ukuran gambar yang digunakan adalah 96 x 96 x 3 (RGB).

1.2.5 Convolutional Neural Network (Covnet)

Convolutional Neural Network adalah salah satu algoritma yang digunakan pada data gambar. CNN digunakan untuk melakukan pengenalan pola pada gambar. CNN terdiri atas beberapa lapisan konvolusi yang kemudian diikuti oleh *Fully Connected Lapisan*. CNN secara khusus terinspirasi korteks visual biologis (Martinelli et al., 2017). CNN merupakan inovasi yang paling berpengaruh pada ranah visi komputer. Metode ini memiliki kemampuan yang lebih baik dibandingkan dengan metode tradisional visi komputer dan telah menciptakan hasil yang mutakhir (Yang et al., 2018). Jaringan syaraf ini telah berhasil diterapkan pada beberapa kasus kehidupan dan aplikasi , seperti klasifikasi gambar, dekteksi objek, segmentasi dan face recognition.

Jaringan syaraf menerima masukan dan mengubahnya melalui serangkaian lapisan tersembunyi (*hidden lapisan*). Setial lapisan tersembunyi tersusun atas satu set neuron diaman seluruh neuron terhubung dengan neuron pada lapisan sebelumnya. Dan pada akhir jaringan terdapat lapisan keluaran yang menghasilkan skor dari klasifikasi (Bakanovskaya et al., 2016). Pada lapisan masukan, CNN terdiri atas 3 masukan yaitu lebar ,tinggi , dan kedalaman dari citra.

Arsitektur covnet merupakan kasus sederhana yang terdiri dari beberapa lapisan yang mentransformasikan volume gambar menjadi volume keluaran. Covnet terdiri atas beberpa jenis lapisan yang berbeda, yakni lapisan konvolusi, lapisan *fully connected* ,fungsi aktivasi, dan lapisan *pooling* namun tidak semua lapisan memiliki nilai parameter.

Berikut beberapa arsitektur yang telah ada, diantaranya adalah :

- a. LeNet. Merupakan arsitektur pertama yang telah berhasil yang disusun oleh Yann LeCun pada tahun 1990. Dengan ini, arsitektur leNet digunakan untuk membaca kode zip, digit, dan lainnya.

- b. AlexNet. Salah satu arsitektur yang populer adalah Alexnet yang dikembangkan oleh Alex Krizhevsky, Ilya Sutskever dan Geoff Hinton. AlexNet telah diajukan dalam tantangan ImageNet ILSVRC pada tahun 2012. Jaringan sangat mirip dengan LeNet, tetapi lebih besar, dan menampilkan lapisan konvolusi yang bertumpuk (Jing sun et al., 2016).
- c. ZF Net. Merupakan pemenang dari ILSVRC 2013 dari arsitektur *convolutional neural network* dari Matthew Zeiler dan Rob Fergus yang diberi nama ZFNet (kepanjangan dari Zeiler dan Fergus Net) (Du et al., 2018).
- d. GoogLeNet. Merupakan pemenang ILSVRC 2014 dari jaringan *convolutional neural network* dari Szegedy et al. dari google. Ini merupakan kontribusi utama dari pengembangan modul *inception* (Szegedy et al., 2015).
- e. VGGNet. Merupakan lapisan CNN yang disusun oleh Karen Simonyan dan Andrew Zisserman yang diketahui segai jaringan VGGNet. Dalam kontribusinya menunjukkan kedalaman dari jaringan untuk performa yang baik (Qiao et al., 2017).
- f. ResNet. Residual Network yang dikembangkan oleh Kaliming He et al. pemenang dari ILSVRC 2015 (Kang et al., 2017).

1.2.6 Proses Konvolusi

Konvolusi dapat digunakan dalam *image processing* untuk menerapkan operator yang mempunyai nilai output dari *pixel* yang berasal dari kombinasi linear nilai *input pixel* tertentu (Such et al., 2017).

Konvolusi merupakan teknik untuk menghaluskan atau memperjelas citra dengan menggantikan nilai piksel dengan sejumlah nilai piksel yang sesuai atau berdekatan dengan piksel aslinya. Tetapi dengan adanya konvolusi, ukuran dari citra tetap sama, tidak berubah. Tujuan dilakukannya konvolusi adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input yang sesuai informasi spasial pada data. Adapun filter gambar

untuk konvolusi seperti deteksi tepi, *sharpen* dan *blur* dengan mengganti nilai pada matriks konvolusi ditunjukkan oleh gambar 2.2 :

Operation	Filter
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

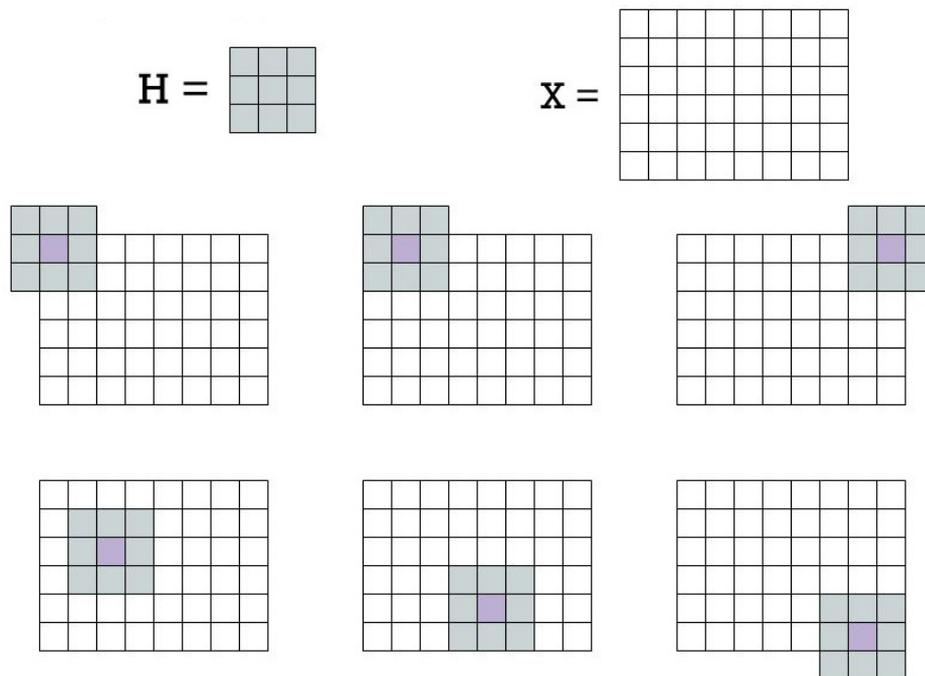
Gambar 2. 2. Matriks Konvolusi

Operasi konvolusi diskrit antara gambar f dan ukuran filter matriks g dirumuskan sebagai berikut :

$$h[x, y] = f[x, y] * g[x, y] = \sum_n \sum_m f[n, m]g[x - n, y - m] \quad (1)$$

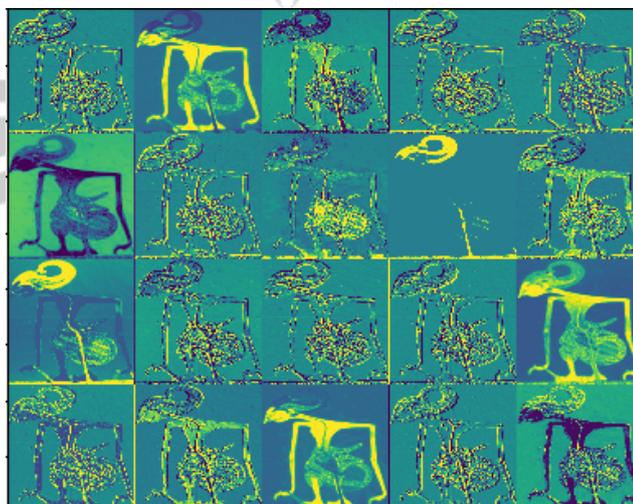
Hasilnya, filter g dan sub gambar f pada koordinat yang sama x, y menghasilkan nilai pixel yang baru pada h pada koordinat x, y . Pada kasus jaringan syaraf, keluaran dari matriks dinamakan *feature map* atau *activation map* apabila telah mengalami operasi fungsi aktivasi.

Ilustrasi konvolusi ditunjukkan pada gambar 2.3:



Gambar 2. 3. Proses Konvolusi

Kernel bergerak dari sudut kiri atas ke kanan bawah pada citra yang berukuran 3x3 atau 5x5 kemudian mengekstraksi 3D *patch feature* disekitarnya (bentuk(*window_widht*, *window_height*, *input_depth*)). Hasil dari konvolusi pada lapisan pertama ditunjukkan pada gambar .



Gambar 2. 4. Hasil Konvolusi Lapisan CNN

Dalam penerapannya, CNN mempelajari nilai dari filter tersebut sendiri saat proses training (Harry Pratt et al., 2016). Sehingga pada CNN *user* harus mengatur parameter seperti jumlah filter, ukuran filter, architecture dari jaringan dan lainnya. Ukuran dari *Feature Map* dikendalikan oleh 3 parameter yang wajib ditentukan sebelum proses konvolusi. Adapun ketiganya adalah:

- a. *Depth*: Merupakan jumlah filter yang digunakan dalam operasi konvolusi.
- b. *Stride*: *Stride* merupakan nilai pixel yang menentukan jarak pergerseran kernel pada gambar, Apabila nilai stride adalah 1, maka kernel bergerak dengan jarak 1 pixel, begitu seterusnya dengan nilai stride yang berbeda.
- c. *Zero Padding*: *Zero padding* memberikan nilai 0 disekitar perbatasan matriks, sehingga nilai filter bisa diterapkan hingga di perbatasan sisi-sisi matriks.

1.2.7 *Machine Learning*

Machine learning atau pembelajaran mesin merupakan pendekatan dalam *artificial intelligence* yang banyak digunakan untuk menirukan perilaku manusia dalam menyelesaikan masalah atau optimasi melakukan optimasi (Lavecchia, 2015). *Machine learning* mencoba untuk menirukan manusia dan menggeneralisasi. Dalam machine learning terdapat dua aplikasi utama yaitu proses pelatihan, pembelajaran ,dan pelatihan. Sehingga dengan adanya hal tersebut, *machine learning* membutuhkan data untuk dipelajari (Dhaoui et al., 2017). Sedangkan Prediksi atau regresi digunakan untuk menentukan data keluaran berdasarkan data pelatihan.

1.2.8 *Deep Learning*

Deep learning juga yang dikenal sebagai *deep structured learning* atau *hierarchical learning* merupakan bagian dari *machine learning* dalam mempelajari representasi data. Pembelajaran dapat secara *supervised*, *semi-supervised* atau *unsupervised*. *Deep learning* arsitektur telah di aplikasikan dalam beberapa bidang seperti *computer vision*, pengenalan suara, *natural language processing*,

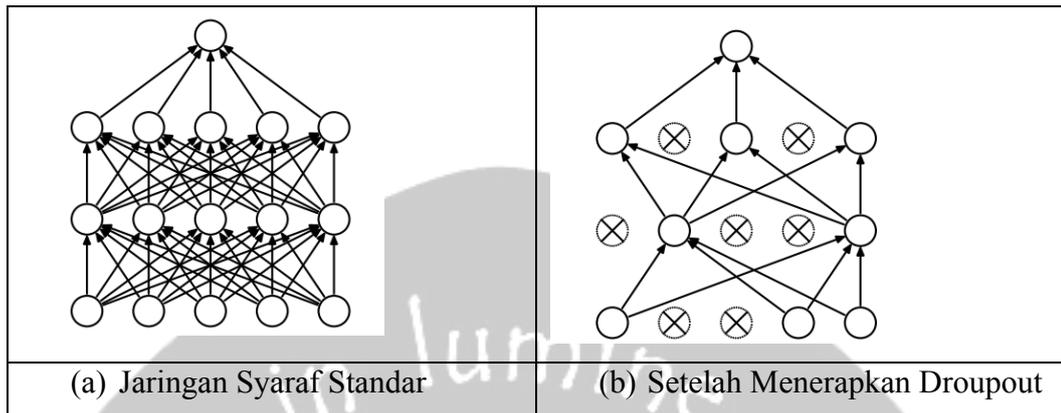
pengenalan suara, filterisasi pada social media, translasi mesin, dan *bioinformatics*. Dimana mereka telah menghasilkan hasil yang sebanding dan unggul dalam beberapa kasus dalam *human expert* (Kim et al., 2016).

1.2.9 Artificial Neural Network

Artificial Neural Network merupakan bagian dari *machine learning* yang menirukan kinerja otak manusia. *Neural Network* terdiri dari lapisan masukan (*input lapisan*) dan keluaran (*output lapisan*). Pada setiap neuron terdapat fungsi aktifasi yang menentukan keluaran unit. Untuk menambah kemampuan neural network dapat ditambahkan jaringan tersembunyi (*hidden lapisan*) (Kavzoglu et al., 2003). Dalam melatih *neural network*, dibutuhkan data pelatihan. Makin banyak data pelatihan yang dimiliki maka kinerja dari neural network semakin bagus. Kemampuan NN tergantung dari jumlah lapisannya. Semakin banyak jumlah lapisan suatu neural network maka semakin tinggi kapasitasnya. Namun, dengan jumlah lapisan yang banyak membutuhkan lebih banyak iterasi atau training yang dibutuhkan. Maka dengan adanya permasalahan ini dikembangkanlah *deep learning*.

1.2.10 Regularisasi dan Data Augmentasi

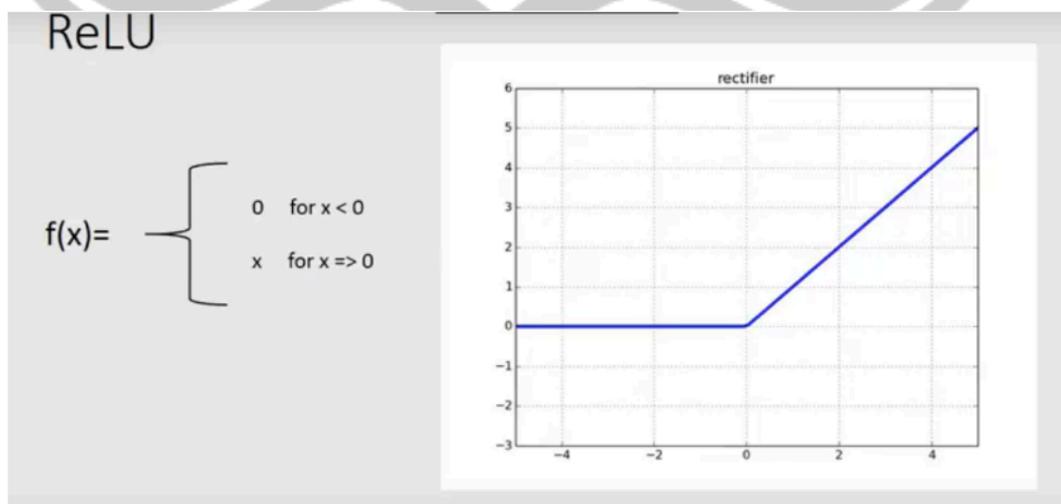
Regularisasi merujuk pada sebuah metode untuk mengurangi *overfitting*. *Overfitting* merupakan suatu kondisi dimana model, dibangun dengan memperhitungkan seluruh ciri yang ada termasuk *noise*. Ada beberapa teknik yang spesifik untuk *deep neural network*, teknik yang populer saat ini adalah *Dropout* (Liang et al., 2015). Istilah *dropout* mengacu pada keluaran unit dalam jaringan syaraf tiruan. Dengan menghilangkan satu unit, yang berarti bahwa menghilangkannya secara sementara dari jaringan bersamaan dengan semua koneksi yang masuk dan keluar, seperti yang ditunjukkan oleh gambar 5.



Gambar 2. 5. Jaringan Syaraf Standar, (b) Contoh Jaringan Syaraf Tiruan yang sudah menerapkan dropout

1.2.11 Fungsi Aktivasi

Fungsi aktivasi merupakan salah satu yang penentu performa dari *neural network* (Mansor et al., 2016). Ada beberapa jenis fungsi aktivasi yang ada, dalam penelitian ini fungsi aktivasi yang digunakan adalah ReLU . ReLU pada dasarnya merupakan sebuah fungsi non-linier yang paling sederhana (Vydana et al., 2017). Apabila mendapat input positif, maka turunannya hanya 1, dengan kata lain aktivasi hanya men-threshold pada nilai 0. Adapun rumus dari ReLU adalah $f(x) = \max(x, 0)$. ReLU digunakan dalam telah digunakan dalam setiap lapisan konvolusi. Output yang diberikan oleh ReLU ditunjukkan seperti gambar 6 :



Gambar 2. 6. Keluaran dari Rectified Linear Unit (ReLU)

ReLU diaplikasikan pada setiap *pixel*, dan mengganti semua nilai *negative pixel* pada *feature map* dengan menggantinya dengan nilai 0.

1.2.12 Pooling Step

Spatial Pooling (juga dapat dikatakan *sub_sampling* atau *down-sampling*) yang berfungsi untuk mengurangi dimensi dari setiap peta fitur tetapi menyimpan informasi yang paling penting. *Spatial Pooling* dapat berbeda bentuk : nilai tertinggi, rata-rata, penjumlahan dan lainnya.

Nilai rata-rata pada *pooling* dapat dirumuskan sebagai berikut :

$$s_i = \frac{1}{n} \sum_{i \in R_j} h_j \quad (2)$$

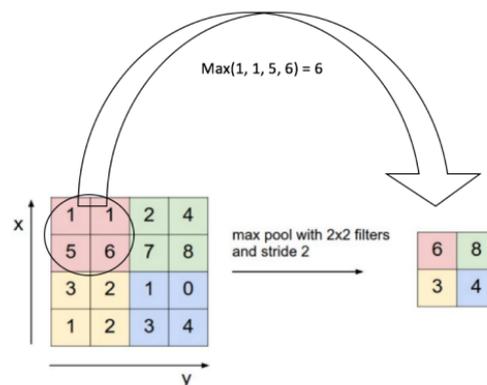
diman h merupakan beberapa piksel pada sub-region R_j dari *feature mapping*, dan n merupakan nomor ciri pada sub-region dimana *pooling* diperlukan.

Max Pooling dirumuskan sebagai berikut :

$$s_i = \max_{i \in R_j} h_i \quad (3)$$

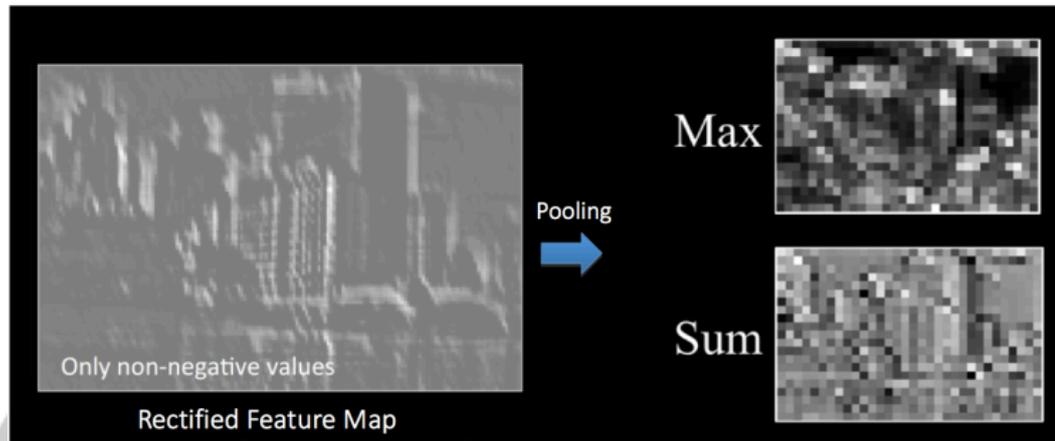
Hasil akhir dari lapisan *pooling* adalah untuk mengurangi ukuran dimensi pada gambar.

Gambar 7 menunjukkan contoh operasi *max pooling* dari hasil setelah operasi ReLU menggunakan 2×2 filter.



Gambar 2. 7. Operasi Max Pooling

Gambar 8 menunjukkan hasil dari *pooling* pada *Rectified Feature Map*



Gambar 2. 8. Hasil dari lapisan pooling

Adapun manfaat dengan adanya *max pooling* adalah sebagai berikut :

- a. Membuat representasi input (dimensi fitur) lebih kecil dan lebih mudah dikelola.
- b. Mengurangi jumlah parameter dan komputasi pada jaringan. Sekaligus untuk mengendalikan *overfitting*.

1.2.13 Fully Connected Lapisan

Fully Connected Lapisan merupakan tradisional *multi lapisan perceptron* yang menggunakan fungsi aktivasi *softmax* pada output layer. Kata “*Fully Connected Lapisan*” mengartikan bahwa setiap neuron di lapisan sebelumnya sebelumnya terhubung ke setiap neuron pada lapisan berikutnya.

Jumlah probabilitas dari keluaran *Fully Connected Lapisan* adalah 1. Ini disebabkan oleh aktivasi *softmax* yang berada pada *Fully Connected Lapisan*. *Softmax* mengambil vector secara acak dan merangkumnya ke dalam suatu vector dengan nilai dari 0 hingga 1 yang berjumlah 1.

Sehingga dengan menggabungkan seluruh proses, dimana Koneksi + *pooling* bertindak sebagai ekstraksi fitur dari gambar masukan dan *Fully Connected Lapisan* bertindak sebagai *classifier*.

Proses pembelajaran secara keseluruhan dari *Convolutional Neural Network* dirangkum sebagai berikut :

- a. Step 1 : Menginisialisasi seluruh filter kernel dan parameter dengan nilai acak.
- b. Step 2 : Jaringan mengambil gambar untuk pelatihan sebagai input, melalui langkah *forward propagation* (operasi konvolusi, ReLU, dan *pooling* secara bersamaan dengan propagasi maju pada *fully connected lapisan*) dan menemukan probabilitas keluaran untuk setiap class.
- c. Step 3: menghitung total *error* pada lapisan output :
$$\text{Total error} = \sum \frac{1}{2} (\text{target} - \text{output})^2$$
- d. Step 4: Menggunakan backpropagation menghitung *gradient error* dan menggunakan *gradient descent* untuk memperbaharui seluruh nilai filter / bobot untuk meminimalkan nilai kesalahan output.
- e. Step 5: mengulangi langkah 2 – 4 dengan seluruh gambar pada data pelatihan.

1.2.14 *Precision, Recall* dan Akurasi

Precision merupakan tingkat ketepatan antara informasi yang diminta, oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan *recall* merupakan tingkat keberhasilan suatu sistem dalam menentukan kembali sebuah informasi.

Pengukuran dalam penelitian ini juga dilihat dari nilai akurasinya. Akurasi merupakan tingkat kedekatan antara nilai prediksi dengan nilai actual.

Secara umum *precision* dirumuskan sebagai berikut :

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall dirumuskan sebagai berikut :

$$\text{Recall} = \frac{TP}{TP + FN}$$

Sedangkan akurasi dirumuskan sebagai berikut :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Keterangan :

1. TP (*True Positive*) merupakan suatu keluaran dimana model dapat dengan benar memprediksi kelas positif.
2. TN (*True Negative*) merupakan suatu keluaran dimana model dapat dengan benar memprediksi kelas negatif.
3. FP (*False Positive*) merupakan suatu keluaran dimana model tidak dapat memprediksi kelas positif.
4. FN (*False Negative*) merupakan suatu keluaran dimana model tidak dapat memprediksi kelas negatif.