

**KAJIAN AKURASI MODEL *ENSEMBLE CONVOLUTIONAL
NEURAL NETWORK* UNTUK KLASIFIKASI PENYAKIT
TANAMAN TOMAT BERDASARKAN DATA CITRA DAUN**

SKRIPSI



RIO MUKHTAROM

H13116011

PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

JANUARI 2020



**KAJIAN AKURASI MODEL *ENSEMBLE*
CONVOLUTIONAL NEURAL NETWORK UNTUK
KLASIFIKASI PENYAKIT TANAMAN TOMAT
BERDASARKAN DATA CITRA DAUN**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains
pada Program Studi Ilmu Komputer Departemen Matematika Fakultas
Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

RIO MUKHTAROM

H13116011

**PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

UNIVERSITAS HASANUDDIN

MAKASSAR

JANUARI 2020

ii



LEMBAR PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul:

KAJIAN AKURASI MODEL *ENSEMBLE CONVOLUTIONAL NEURAL NETWORK* UNTUK KLASIFIKASI PENYAKIT TANAMAN TOMAT BERDASARKAN DATA CITRA DAUN

adalah benar hasil karya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Makassar, 29 Januari 2020



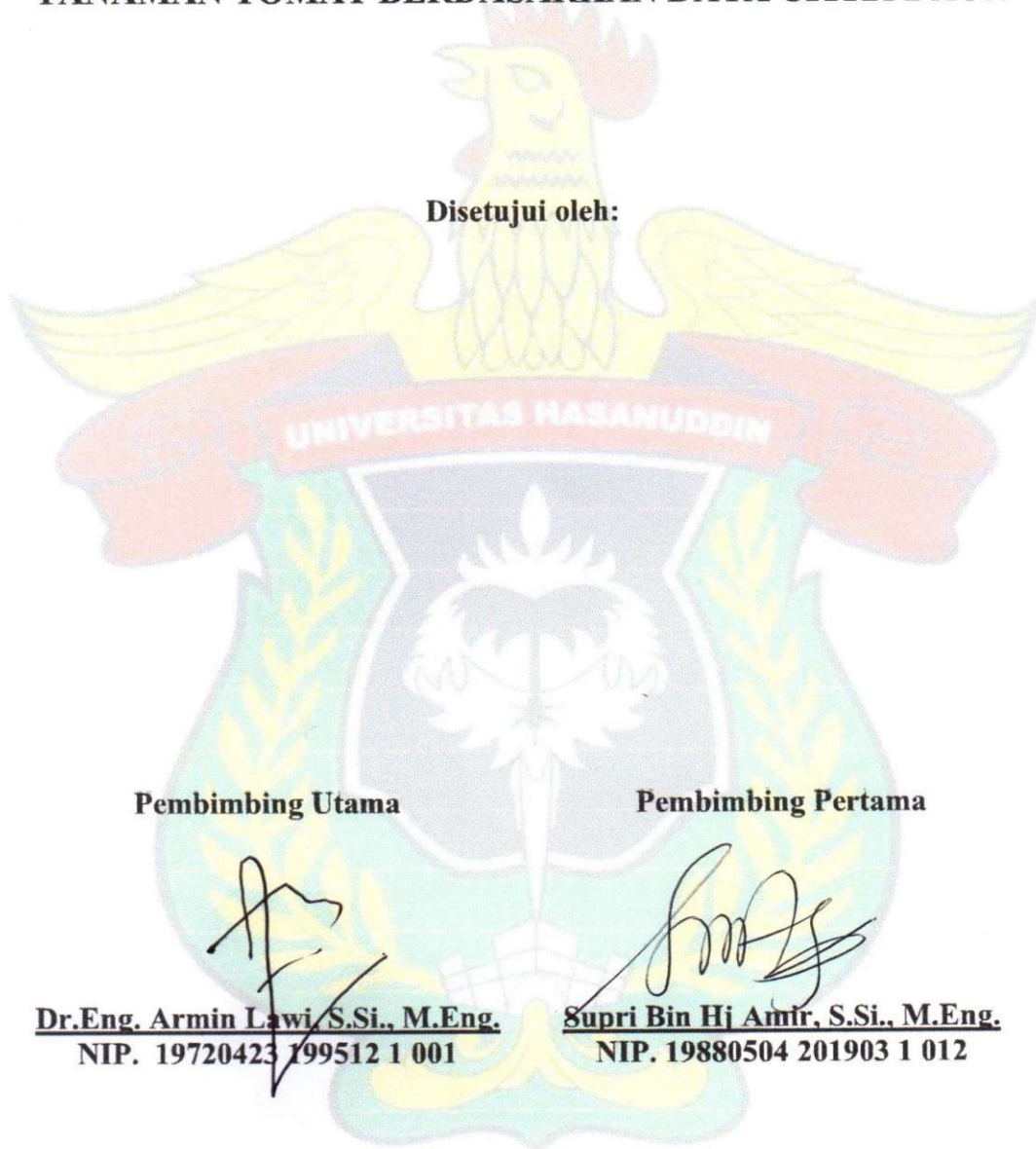
Rio Mukhtarom

NIM. H13116011



KAJIAN AKURASI MODEL *ENSEMBLE CONVOLUTIONAL NEURAL NETWORK* UNTUK KLASIFIKASI PENYAKIT TANAMAN TOMAT BERDASARKAN DATA CITRA DAUN

Disetujui oleh:



Pembimbing Utama

Pembimbing Pertama

Dr.Eng. Armin Lawi, S.Si., M.Eng.
NIP. 19720423 199512 1 001

Supri Bin Hj Amir, S.Si., M.Eng.
NIP. 19880504 201903 1 012

Pada 29 Januari 2020



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Rio Mukhtarom
NIM : H13116011
Program Studi : Ilmu Komputer
Judul Skripsi : Kajian Akurasi Model *Ensemble Convolutional Neural Network* untuk Klasifikasi Penyakit Tanaman Tomat berdasarkan Data Citra Daun

Telah berhasil mempertahankan di hadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

1. Ketua : Dr.Eng. Armin Lawi, S.Si., M.Eng.
2. Sekretaris : Supri Bin Hj. Amir, S.Si., M.Eng.
3. Anggota : Dr. Hendra, S.Si., M.Kom.
4. Anggota : Edy Saputra, S.Si., M.Si.

Tanda Tangan

(.....)
(.....)
(.....)
(.....)

Ditetapkan di : Makassar

Tanggal : 29 Januari 2020



KATA PENGANTAR

Segala puji bagi Allah *Subhanahu Wa ta'ala*, Tuhan alam semesta yang telah memberikan nikmat kesempatan, kesehatan dan kemampuan sehingga penulisan skripsi ini bisa selesai. Shalawat serta salam senantiasa tercurah kepada *Rasulullah* Muhammad *Shallallahu Alaihi Wasallam*, yang merupakan teladan dalam menjalankan kehidupan di dunia.

Alhamdulillah, skripsi dengan Judul “Kajian Akurasi Model *Ensemble Convolutional Neural Network* untuk Klasifikasi Penyakit Tanaman Tomat berdasarkan Data Citra Daun” yang disusun sebagai salah satu syarat akademik untuk meraih gelar Sarjana Sains pada Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin ini dapat diselesaikan. Tentunya, dalam penulisan skripsi ini, penulis mampu menyelesaikan tepat waktu berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, ucapan terima kasih dan apresiasi yang tak terhingga kepada kedua orang tua penulis, Ayahanda **Paweroi** dan Ibunda **Musdalipah** yang tak kenal lelah dalam memanjatkan doa serta memberikan nasihat dan motivasi kepada penulis. Tugas akhir ini hanya setitik kebahagiaan kecil yang bisa penulis persembahkan. Tidak lupa pula terima kasih kepada saudara-saudara saya, **Vivi Feromida**, **Awii Musawwir**, dan **Muro Mukromina**, yang senantiasa memberikan dukungan dan semangat.

Terima kasih juga penulis ucapkan kepada:

1. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu** beserta jajarannya.
2. Dekan Fakultas Matematika dan Ilmu Pengetahuna Alam (FMIPA), **Dr.Eng. Amiruddin** beserta jajarannya.
3. Bapak **Dr. Nurdin, S.Si., M.Si.** sebagai Ketua Departemen Matematika FMIPA Unhas, bapak **Dr. Diaraya, M.Ak.** sebagai Ketua Program Studi Ilmu Komputer Unhas, dosen-dosen pengajar, dan staf Departemen Matematika atas dan bantuan yang selama ini telah diberikan.



4. Bapak **Dr.Eng. Armin Lawi, S.Si., M.Eng.** sebagai dosen pembimbing utama sekaligus ketua tim penguji atas semua ilmu yang telah diberikan selama proses perkuliahan dan senantiasa memotivasi penulis dalam penulisan skripsi ini.
5. Bapak **Supri Bin Hj. Amir, S.Si., M.Eng.** sebagai dosen pembimbing pertama sekaligus sekteraris tim penguji atas ilmu yang diberikan selama proses perkuliahan dan bimbingan, serta segala bentuk bantuan yang telah diberikan dalam penyusunan skripsi ini.
6. Bapak **Dr. Hendra, S.Si., M.Kom.** sebagai anggota tim penguji atas segala ilmu yang telah diberikan selama proses perkuliahan serta berbagai masukan dan kritik yang membangun dalam proses penyusunan skripsi ini.
7. Bapak **Edy Saputra, S.Si., M.Si.** sebagai anggota tim penguji atas segala kritikan dan masukan yang membangun dalam penyusunan skripsi ini.
8. Bapak **Andi Galsan Mahie, S.Si., M.Si.** sebagai dosen pembimbing akademik yang senantiasa memberikan motivasi, dorongan, dan masukan dalam hal akademik selama menjadi mahasiswa Ilmu Komputer Unhas.
9. Saudara **ILMU KOMPUTER UNHAS 2016** atas kebersamaan, kepedulian, suka duka, canda tawa yang telah dilalui selama ini. Semoga persahabatan dan kebersamaan kita tidak habis dimakan waktu.
10. Saudara **SSC Squad** yang senantiasa ada untuk memberikan masukan dan solusi terhadap masalah yang dihadapi penulis.
11. Keluarga besar **HIMATIKA FMIPA UNHAS** terkhusus **ALGORITMA 2016** atas segala bentuk dukungan dan bantuan selama menjalani kehidupan kampus. Semoga kesuksesan selalu kita dapatkan dalam setiap langkah kita.
12. Keluarga besar **KM FMIPA UNHAS** terkhusus kepada **MIPA 2016** atas persahabatan, kekerabatan, kerjasama, serta cerita-cerita lain yang telah kita ukur bersama.
13. Keluarga besar **UKM KPI UNHAS**, terkhusus **Angkatan 9 UKM KPI UNHAS** dan **Jajaran Kabinet Semangat Bersatu 2019** atas segala cerita, pengalaman, dan pelajaran selama mengemban amanah. Terima kasih telah

adi rumah pengembangan skill penulis.

uarga besar **LDM AL-AQSO UNHAS** terkhusus kepada **MADU 2016** yang
menjadi penyemangat ketika menghadapi masalah hidup.



15. Teman-teman **KKN E-Commerce Luwu Utara** yang telah bersama menjalani pengabdian untuk UMKM yang ada di Kabupaten Luwu Utara.
16. Semua pihak yang tidak dapat disebutkan satu persatu, atas segala bentuk kontribusi, partisipasi, serta motivasi yang diberikan kepada penulis selama ini. Semoga apa yang telah diberikan akan dilipatgandakan oleh Allah.

Penulis menyadari bahwa masih banyak kekurangan dalam tugas kahir ini, untuk itu, dengan segala kerendahan hati penulis mohon maaf sebesar-besarnya. Semoga tulisan ini memberikan manfaat untuk siapapun yang membacanya.

Makassar, 29 Januari 2020

Rio Mukhtarom



PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Rio Mukhtarom
NIM : H13116011
Programa Studi : Ilmu Komputer
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas tugas akhir saya yang berjudul:

**“Kajian Akurasi Model *Ensemble Convolutional Neural Network* untuk
Klasifikasi Penyakit Tanaman Tomat berdasarkan Data Citra Daun”**

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada 29 Januari 2020

Yang menyatakan



(Rio Mukhtarom)



ABSTRAK

Machine learning merupakan metode yang digunakan untuk menganalisis masalah dalam *data mining*. Salah satu masalah yang bisa diselesaikan melalui metode *machine learning* adalah klasifikasi. Pada masalah klasifikasi menggunakan data citra, algoritma yang digunakan adalah *Convolutional Neural Network (CNN)*. Dalam perkembangannya, penelitian tentang CNN banyak mengarah pada tingkat kedalaman struktur *network (Deep Neural Network)*. Namun di sisi lain, perkembangan metode *ensemble* dengan mengombinasikan banyak algoritma juga semakin berkembang. Maka dari itu, penelitian ini menerapkan metode *ensemble* pada CNN untuk klasifikasi penyakit tanaman tomat berdasarkan data citra daun. Enam model CNN dengan arsitektur yang berbeda dilakukan *training*. Model hasil *training* kemudian dikombinasikan menggunakan metode *ensemble-adaboost* untuk mendapatkan bobot masing-masing model. Bobot dikalikan dengan hasil output dari masing-masing model untuk mendapatkan output hasil *ensemble*. Hasilnya, akurasi data *training ensemble* mencapai 0,9883. Akurasi ini secara umum lebih tinggi dibanding model-model yang lain.

Kata Kunci: *Convolutional Neural Network (CNN)*, *Ensemble*, *Machine Learning*, Klasifikasi



ABSTRACT

Machine learning is a method used to analyze problems in data mining. One of the problems that can be solved by machine learning is classification. In the classification problem using image data, the algorithm used is Convolutional Neural Network (CNN). In its development, studies about CNN lead to the level of depth of network structure (Deep Neural Network). But on the other hand, the development of ensemble methods by combining many algorithms is also growing. Therefore, this research applies the ensemble method on CNN for the classification of tomato plant diseases based on leaf image data. Six CNN models with different architectures were training. The training result models are then combined using the ensemble-adaboost method to get the weight of each model. The weight is multiplied by the output of each model to get the ensemble output. The result, the accuracy of the ensemble training data reached 0.9883. This accuracy is generally higher than other models.

Keywords: Convolutional Neural Network (CNN), Ensemble, Machine Learning, Classification



DAFTAR ISI

HALAMAN JUDUL.....	ii
LEMBAR PERNYATAAN KEOTENTIKAN.....	iii
PERSETUJUAN PEMBIMBING.....	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	ix
ABSTRAK	x
ABSTRACT	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Landasan Teori	5
2.2 Penelitian Terkait	17
METODOLOGI PENELITIAN.....	20
Tahapan Penelitian	20
	xii



3.2	Waktu dan Lokasi Penelitian.....	22
3.3	<i>Dataset</i>	22
3.4	Instrumen Penelitian.....	22
BAB IV HASIL DAN PEMBAHASAN		23
4.1	Deskripsi Data	23
4.2	<i>Preprocessing</i>	24
4.3	Implementasi Arsitektur CNN.....	26
4.4	<i>Training Data</i>	28
4.5	<i>Ensemble</i>	33
4.6	Analisis Kinerja.....	35
BAB V KESIMPULAN DAN SARAN.....		41
5.1	Kesimpulan.....	41
5.2	Saran.....	41
DAFTAR PUSTAKA		42
LAMPIRAN.....		44



DAFTAR GAMBAR

Gambar 1 Ilustrasi Representasi citra <i>grayscale</i> ukuran 3x3.....	7
Gambar 2 Representasi citra RGB ukuran 3x3	7
Gambar 3 Contoh arsitektur CNN.....	8
Gambar 4 Ilustrasi operasi konvolusi.....	9
Gambar 5 Ilustrasi operasi konvolusi pada citra RGB.....	10
Gambar 6 Ilustrasi operasi konvolusi pada citra RGB dengan layer <i>filter</i> lebih dari satu	10
Gambar 7 Grafik fungsi ReLU.....	11
Gambar 8 Ilustrasi operasi <i>pooling</i>	12
Gambar 9 Ilustrasi <i>flattening</i>	13
Gambar 10 <i>Fully connected layer</i>	13
Gambar 11 Struktur <i>ensemble learning</i>	14
Gambar 12 Kurva ROC.....	17
Gambar 13 Diagram alur penelitian.....	20
Gambar 14 Arsitektur CNN	21
Gambar 15 Ilustrasi <i>resize</i>	24
Gambar 16 Ilustrasi representasi citra.....	25
Gambar 17 Ilustrasi <i>rescale</i>	26
Gambar 18 Arsitektur CNN	27
Gambar 19 Arsitektur CNN 1	28
Gambar 20 Grafik perubahan akurasi terhadap jumlah <i>epoch</i> pada data <i>train</i> dan validasi CNN 1	28
Gambar 21 Arsitektur CNN 2	29
Gambar 22 Grafik perubahan akurasi terhadap jumlah <i>epoch</i> pada data <i>train</i> dan validasi CNN 2	29
Gambar 23 Arsitektur CNN 3	30
Gambar 24 Grafik perubahan akurasi terhadap jumlah <i>epoch</i> pada data <i>train</i> dan validasi CNN 3	30
Gambar 25 Arsitektur CNN 4	30



Gambar 26 Grafik perubahan akurasi terhadap jumlah <i>epoch</i> pada data <i>train</i> dan validasi CNN 4	31
Gambar 27 Arsitektur CNN 5	31
Gambar 28 Grafik perubahan akurasi terhadap jumlah <i>epoch</i> pada data <i>train</i> dan validasi CNN 5	32
Gambar 29 Arsitektur CNN 6	32
Gambar 30 Grafik perubahan akurasi terhadap jumlah <i>epoch</i> pada data <i>train</i> dan validasi CNN 6	33
Gambar 31 Grafik perbandingan akurasi masing-masing model CNN dengan <i>ensemble</i> CNN.....	34
Gambar 32 Grafik Perbandingan Akurasi Model	35
Gambar 33 Grafik Perbandingan Waktu Komputasi	35
Gambar 34 Nilai <i>Precision</i>	36
Gambar 35 Nilai <i>Recall</i>	36
Gambar 36 Kurva ROC setiap kelas pada model CNN 1	38
Gambar 37 Kurva ROC setiap kelas pada model CNN 2	38
Gambar 38 Kurva ROC setiap kelas pada model CNN 3	38
Gambar 39 Kurva ROC setiap kelas pada model CNN 4	39
Gambar 40 Kurva ROC setiap kelas pada model CNN 5	39
Gambar 41 Kurva ROC setiap kelas pada model CNN 6	39
Gambar 42 Kurva ROC setiap kelas pada model <i>ensemble</i>	40
Gambar 43 <i>Confusion matrix</i> evaluasi model CNN 1	47
Gambar 44 <i>Confusion matrix</i> evaluasi model CNN 2	48
Gambar 45 <i>Confusion matrix</i> evaluasi model CNN 3	49
Gambar 46 <i>Confusion matrix</i> evaluasi model CNN 4	50
Gambar 47 <i>Confusion matrix</i> evaluasi model CNN 5	51
Gambar 48 <i>Confusion matrix</i> evaluasi model CNN 6	52
Gambar 49 <i>Confusion matrix</i> evaluasi model <i>ensemble</i>	53



DAFTAR TABEL

Tabel 1 Jenis Penyakit Tanaman Tomat	6
Tabel 2 Data citra penyakit daun tomat	23
Tabel 3 Daftar pembagian data	26
Tabel 4 Ukuran Kernel Konvolusi Setiap Model CNN	28
Tabel 5 Bobot masing-masing model	33
Tabel 6 Nilai Precision dan Recall	54
Tabel 7 Akurasi dan waktu komputasi.....	54



DAFTAR LAMPIRAN

Lampiran 1 Souce Code	44
Lampiran 2 Performa Model CNN 1	47
Lampiran 3 Performa Model CNN 2	48
Lampiran 4 Performa Model CNN 3	49
Lampiran 5 Performa Model CNN 4	50
Lampiran 6 Performa Model CNN 5	51
Lampiran 7 Performa Model CNN 6	52
Lampiran 8 Performa <i>Ensemble</i>	53
Lampiran 9 Nilai Precision dan Recall data test	54
Lampiran 10 Akurasi dan waktu komputasi data test	54



BAB I

PENDAHULUAN

1.1 Latar Belakang

Machine learning merupakan salah satu bagian disiplin ilmu dari *data mining*. *Machine learning* merupakan bagian dalam kecerdasan buatan yang berhubungan dengan pengembangan teknik-teknik yang bisa diprogramkan dan belajar dari data masa lalu. *Machine learning* juga merupakan metode yang digunakan untuk melakukan analisis masalah dalam *data mining*. Salah satu masalah dasar yang dapat diselesaikan melalui metode *machine learning* adalah klasifikasi (Santosa & Umam, 2018).

Beberapa algoritma *machine learning* yang sering digunakan dalam melakukan klasifikasi yaitu *K-NearestNeighbor (KNN)*, *Decision Tree*, dan *Naïve Bayesian* dan masih banyak lagi. Algoritma tersebut dapat memperoleh tingkat akurasi klasifikasi yang baik pada umumnya. Namun, algoritma tersebut masih memiliki beberapa kekurangan. Pertama, algoritma tersebut membutuhkan fitur buatan dari hasil ekstraksi yang dapat menghasilkan ketidakpastian fitur. Kedua, algoritma klasifikasi ini tidak dapat menghasilkan akurasi klasifikasi yang sangat tinggi. Kemudian, mulai berkembang metode *ensemble* yang menggabungkan beberapa model klasifikasi untuk meningkatkan akurasinya. Namun, karena dibatasi oleh model klasifikasi yang akurasinya rendah, model *ensemble* masih tidak bisa mendapatkan akurasi yang sangat tinggi, meskipun akurasinya dapat meningkat (Yang, Chen, Yan, Zhao, & Fan, 2017).

Artificial Neural Network (ANN) merupakan algoritma klasifikasi yang terinspirasi dari sistem jaringan saraf makhluk hidup. Jaringan tersebut terdiri dari neuron yang sangat banyak dan saling terhubung (Santosa & Umam, 2018). Dalam perkembangannya, ANN menjadi algoritma klasifikasi yang dapat menghasilkan akurasi yang tinggi. Salah satu pengembangan dari ANN yaitu *Convolutional Neural Network (CNN)*. CNN banyak digunakan untuk mengolah data citra. Salah

sektor CNN yang diberi nama LeNet dapat menghasilkan akurasi 98% (Botton, Bengio, & Haffner, 1998). Namun, karena keterbatasan perangkat keras pada waktu itu (1998), proses komputasi CNN membutuhkan



banyak waktu dan terjadi *overfitting*. CNN kemudian mulai ditinggalkan karena dianggap tidak efektif. (Yang, Chen, Yan, Zhao, & Fan, 2017)

Hingga pada 2012, Prof. Hinton mendesain arsitektur CNN untuk klasifikasi citra RGB. Arsitektur tersebut diberi nama AlexNet. AlexNet menggunakan lapisan baru seperti ReLU, Dropout, dan normalisasi LRN untuk meningkatkan akurasi (Krizhevsky, Sutskever, & Hinton, 2012). Tahun 2012 ini menjadi masa awal perkembangan pesat dari algoritma CNN.

Pada tahun-tahun selanjutnya, penelitian tentang klasifikasi dengan CNN secara umum mengarah ke kedalaman struktur *network* (*Deep Neural Network*) (Yang, Chen, Yan, Zhao, & Fan, 2017). Pada 2014, Simonyan mendesain arsitektur CNN dengan 16 lapisan (13 lapisan konvolusi dan 3 lapisan *full connected*). Desain arsitektur ini diberi nama *Visual Geometry Group* (VGG). Bahkan ada desain arsitektur dengan lapisan yang lebih banyak, yaitu VGG-19 dengan 19 lapisan (Simonyan & Zisserman, 2014). Pada tahun yang sama, Tim Google mendesain model *network* yang diberi nama GooLenet dengan 22 lapisan (Szegedy, et al., 2014). Pada tahun selanjutnya, tim Microsoft mendesain arsitektur CNN dengan jumlah lapisan yang lebih banyak, hingga 152 lapisan (He, Zhang, Ren, & Sun, 2016).

Meskipun perkembangan *deep neural network* semakin cepat, metode *ensemble* untuk menggabungkan metode pelatihan atau model yang berbeda juga tetap berkembang. Salah satunya yaitu metode *ensemble learning* (*Bagging*) dengan algoritma SVM dengan nama SE-SVM. SE-SVM memilih beberapa jenis SVM dan menggabungkan dengan metode *ensemble*. Hasilnya ternyata efektif dalam meningkatkan performa (Yang, Chen, Yan, Zhao, & Fan, 2017).

CNN pada umumnya sudah di desain dengan lapisan *network* yang dalam sehingga bisa menghasilkan akurasi yang tinggi. Namun, dalam implementasinya membutuhkan sumber daya perangkat keras yang besar juga. Diharapkan dengan adanya algoritma CNN dengan akurasi yang tinggi dikombinasikan dengan metode *ensemble learning* diharapkan dapat menghasilkan model yang sangat baik.

model dapat semakin meningkat serta *error rate* semakin berkurang.

Sejak dari itu, penulis tertarik untuk melakukan penelitian dengan menerapkan metode *ensemble learning* dengan menggabungkan banyak



arsitektur CNN. Penelitian ini berjudul “**Kajian Akurasi Model *Ensemble Convolutional Neural Network* untuk Klasifikasi Penyakit Tanaman Tomat berdasarkan Data Citra Daun**”.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana akurasi dari model *ensemble convolutional neural network* untuk data citra daun tomat?
2. Bagaimana perbandingan akurasi antara *ensemble convolutional neural network* dengan *convolutional neural network* saja untuk data citra daun tomat?

1.3 Batasan Masalah

Berikut ini merupakan beberapa batasan dalam penelitian ini.

1. Algoritma CNN yang diterapkan menggunakan arsitektur berbeda dan akan dikombinasikan menggunakan metode *ensemble*. Perbedaan masing-masing arsitektur terletak pada ukuran layer konvolusinya.
2. Metode *ensemble* yang digunakan yaitu *Adaptive Boosting* (AdaBoost).
3. Identifikasi penyakit dilakukan pada daun tanaman tomat.
4. Jenis penyakit tanaman tomat yang akan diteliti ada 6, yaitu *bacterial spot*, *early blight*, *late blight*, *leaf mold*, *septoria leaf spot*, dan *yellow leaf curl virus*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

1. Menganalisis akurasi yang dihasilkan dari model *ensemble convolutional neural network* untuk data citra daun tomat.
2. Menganalisis perbandingan akurasi antara *ensemble convolutional neural network* dengan *convolutional neural network* saja untuk data citra daun tomat.

1.5 Manfaat Penelitian

Penelitian ini dapat menghasilkan model machine learning yang bisa diintegrasikan kemudian pada alat pendeteksi penyakit tanaman. Selain itu,



penelitian ini dapat menjadi referensi tentang cara menyelesaikan masalah klasifikasi menggunakan metode *Ensemble Convolutional Neural Network*.



BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Penyakit Tanaman Tomat

Tanaman tomat merupakan salah satu tanaman hortikultura yang penting di Indonesia. Namun, budidaya tanaman tomat banyak mengalami masalah yang dapat menyebabkan produksi tanaman tomat menjadi rendah, baik secara kualitas maupun kuantitas. Berdasarkan data Badan Pusat Statistik dan Direktorat Jenderal Hortikultura, produksi tomat dari tahun 2014 ke tahun 2015 menurun 4.16%. Penurunan tersebut disebabkan karena beberapa masalah. Salah satu masalah tersebut merupakan penyakit yang menyerang daun tanaman tomat. Pada penelitian ini, identifikasi terhadap penyakit tanaman tomat dilakukan terhadap daun. Penyakit tanaman tomat yang dapat diidentifikasi dari daun yaitu sebagai berikut (Blake, Keinath, & Kluepfel, 2018).

2.1.1.1 *Bacterial Spot* (Bercak Bakteri)

Penyakit ini disebabkan oleh bakteri *Xanthomonas vesicatoria*. Bakteri ini menyerang buah, daun, dan batang tanaman tomat. Penyakit ini banyak terjadi pada musim hujan. Penyakit ini ditandai dengan kerusakan pada daun yang menimbulkan bercak. Jika daun terserang penyakit ini, akan banyak bintik-bintik kecil, bagian daun akan mengeriting hingga mengering.

2.1.1.2 *Early Blight* (Bercak Daun)

Penyakit ini disebabkan oleh jamur *Alternaria tomatophila* dan *Alternaria Solani*. Jamur ini biasa menyerang pada musim hujan. Penyakit ini bisa diidentifikasi melalui daun dengan melihat bercak kecil berwarna coklat yang dapat meluas dan menyebabkan daun berlubang.

2.1.1.3 *Late Blight* (Busuk Daun)

Penyakit ini ditandai dengan bercak coklat hingga hitam. Bercak akan muncul di ujung, kemudian akan meluas ke seluruh daun. Penyakit ini disebabkan oleh jamur *Phytophthora infestan*. Penyebaran virus ini sangat aktif ketika musim



hujan, terutama dalam keadaan dingin dan lembab. Biasanya penyakit ini menyerang pada tanaman tomat di dataran tinggi.

2.1.1.4 Leaf Mold (Jamur Daun)

Jamur *Passalora fulva* merupakan penyebab dari penyakit ini. Gejala awalnya pada daun berupa bintik-bintik hijau pucat atau kekuningan pada bagian atas. Perlahan akan membesar dan berubah jadi kuning.







2.1.1.5 Septoria Leaf Spot (Bercak Daun Septoria)

Penyakit yang disebabkan oleh jamur *Septoria lycopersici* ini menyerang daun, tangkai daun, dan batang. Infeksi jamur ini biasanya terjadi pada daun bagian bawah dekat tanah setelah tanaman mulai berbuah. Jika terserang, terdapat bintik-bintik melingkar dengan tepi gelap pada daun yang lebih tua.

2.1.1.6 Yellow Leaf Curl Virus

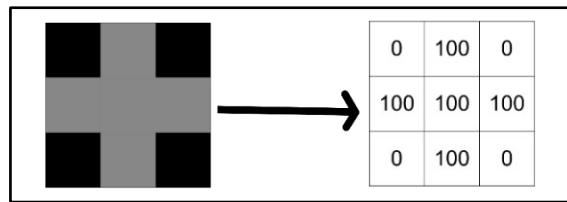
Penyakit ini ditularkan oleh lalat putih. Gejala penyakit ini yaitu daun mengeriting ke atas, menguning, dan lebih kecil dari biasanya. Jika tanaman tomat terinfeksi penyakit ini di awal pertumbuhan, memungkinkan tidak akan ada buah yang terbentuk.

Tabel 1 Jenis Penyakit Tanaman Tomat

Gambar	Keterangan	Gambar	Keterangan
	<i>Bacterial Spot</i>		<i>Septoria Leaf Spot</i>
	<i>Early Blight</i>		<i>Leaf Mold</i>
	<i>Late Blight</i>		<i>Yellow Leaf Curl Virus</i>

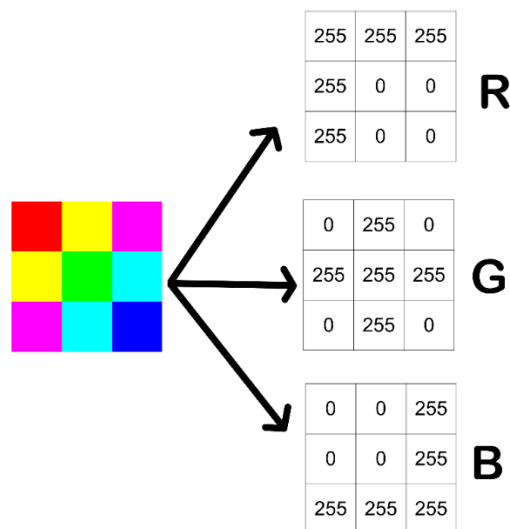


2.1.2 Citra Digital



Gambar 1 Ilustrasi Representasi citra *grayscale* ukuran 3x3

Citra digital merupakan salah satu jenis citra. Citra digital merupakan representasi numerik data gambar agar dapat diolah. Suatu citra digital dapat direpresentasikan dalam bentuk matriks dengan fungsi $f(x,y)$ yang terdiri dari M kolom dan N baris. Perpotongan antara baris dan kolom disebut pixel. Satu pixel mewakili satu warna dan merupakan elemen terkecil dari citra digital (Gonzalez, Woods, & Eddins, 2004). Sebelum data diolah menggunakan *convolutional neural network*, citra terlebih dahulu direpresentasikan ke dalam bentuk numerik (citra digital).



Gambar 2 Representasi citra RGB ukuran 3x3

Pada gambar 1 diilustrasikan hasil representasi citra *grayscale* dengan ukuran 3x3. Hasil representasinya akan dimuat dalam bentuk matriks. Pada kolom pertama baris pertama, warna hitam direpresentasikan sebagai angka 0. Kemudian pada baris kedua kolom kedua warna abu-abu direpresentasikan sebagai angka 100.

Pada gambar 2, diilustrasikan hasil representasi citra RGB dengan ukuran 3x3. Pada citra RGB, hasil representasinya akan terdapat 3 lapisan matriks dengan

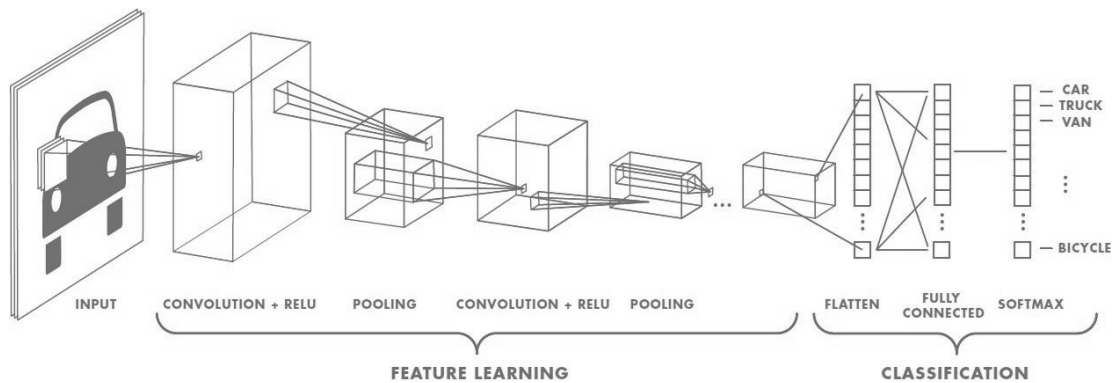


masing-masing matriks merepresentasikan citra *red*, *green*, dan *blue*. Misalnya pada baris ketiga kolom pertama, hasil representasinya yaitu (255, 0, 255).

2.1.3 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu pengembangan dari *Artificial Neural Network* (ANN). CNN merupakan algoritma *neural network* yang digunakan untuk input berupa gambar atau video (Santosa & Umam, 2018). Salah satu keuntungan dari CNN yaitu mengurangi jumlah parameter dalam *Neural Network* (Albawi, Mohammed, & Al-Zawi, 2017).

Secara umum, arsitektur CNN terdiri dari 2 tahap. Pertama, tahap ekstraksi fitur (*feature extraction / feature learning*). Pada tahap ini biasanya mencakup operasi konvolusi, fungsi aktivasi (ReLU), dan *pooling*. Pada tahap ini akan dihasilkan parameter berupa angka-angka hasil representasi dari operasi-operasi yang dilakukan. Kedua, tahap klasifikasi. Hasil dari ekstraksi fitur akan dilakukan klasifikasi melalui *full connected layer*.



Gambar 3 Contoh arsitektur CNN

2.1.3.1 Konvolusi

Konvolusi merupakan salah satu tahap pada arsitektur CNN. Konvolusi merupakan suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Dalam hal pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel pada citra.

umus konvolusi dapat dilihat pada persamaan (1). I merupakan input citra, dan K merupakan *kernel* atau *filter* konvolusi.

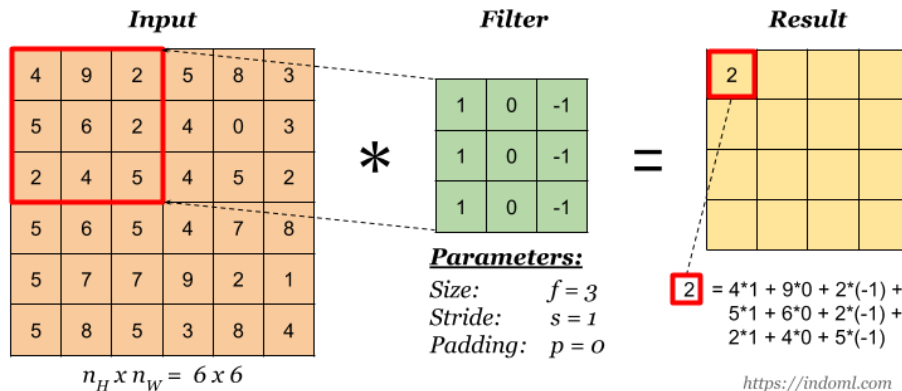


$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i - m, j - n) \quad (1)$$

Keterangan:

- S(i,j) = fungsi hasil operasi konvolusi
- I = input
- K = kernel atau filter
- i,j = pixel input
- m,n = pixel kernel

Tujuan dilakukannya konvolusi pada citra adalah untuk mengekstraksi fitur dari citra *input*. Konvolusi akan menghasilkan transformasi linear dari data *input*. Pada gambar 4 diilustrasikan bagaimana operasi konvolusi pada matriks hasil representasi citra. Matriks input merupakan representasi citra dalam bentuk numerik. Operasi konvolusi akan dilakukan antara matriks *input* ukuran 6x6 dengan matriks *filter* dengan ukuran 3x3. *Stride* atau perpindahan sebanyak satu dengan *zero padding*. Hingga hasilnya seperti pada matriks *result*.

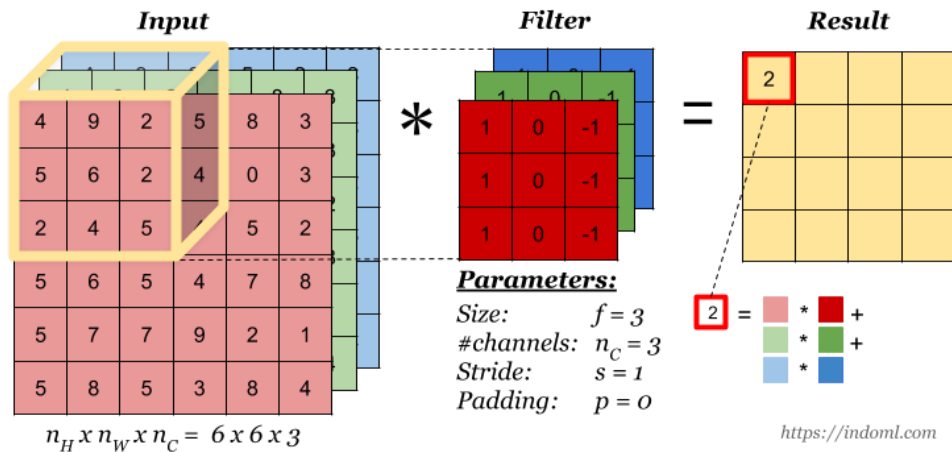


Sumber: indoml.com

Gambar 4 Ilustrasi operasi konvolusi

Operasi konvolusi pada gambar 4 hanya berlaku pada citra *grayscale* yang hasil representasi gambarnya hanya satu lapisan. Untuk citra RGB, representasi gambarnya akan menghasilkan tiga lapisan matriks. Tiga lapisan matriks ini hanya akan menghasilkan satu lapis matriks dari operasi konvolusi. Masing-masing lapisan matriks akan dioperasikan dengan *filter*. Hasil operasi konvolusi dari masing lapisan akan saling dijumlahkan untuk menghasilkan matriks hasil akhirnya seperti pada gambar 5.

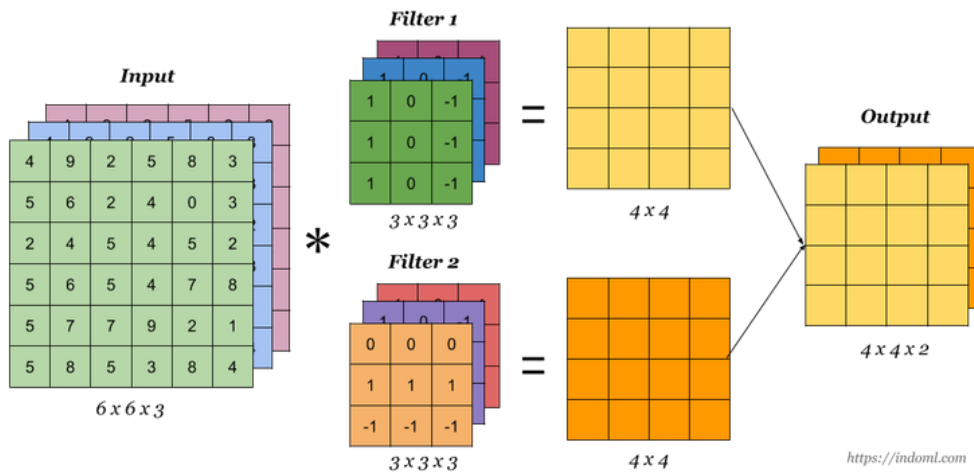




Sumber: indoml.com

Gambar 5 Ilustrasi operasi konvolusi pada citra RGB

Adapun operasi konvolusi pada CNN, biasanya tidak hanya menggunakan satu lapisan *filter*. Pada gambar 6, diilustrasikan bagaimana operasi konvolusi pada citra RGB dengan 2 *filter*. Operasi untuk masing-masing *filter* sama dengan ilustrasi pada gambar 5. Jumlah lapisan output dari hasil operasi konvolusi dengan banyak lapisan sama dengan banyaknya *filter* yang digunakan.



Sumber: indoml.com

Gambar 6 Ilustrasi operasi konvolusi pada citra RGB dengan layer *filter* lebih dari satu

2.1.3.2 Fungsi Aktivasi

Fungsi aktivasi biasa disebut sebagai lapisan pemetaan non-linear. Fungsi digunakan untuk meningkatkan kemampuan klasifikasi *network* (Chen, Wang, 2018). Salah satu peranan dari fungsi aktivasi adalah untuk

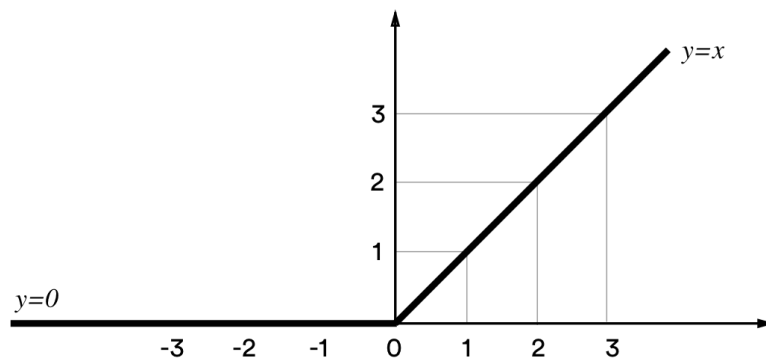


memberikan kemampuan *network* agar dapat melakukan tugas non-linear. Tanpa fungsi aktivasi, *neural network* hanyalah kombinasi operasi linear yang hanya dapat melakukan tugas-tugas yang linear pula. Padahal kebanyakan kasus nyata di lapangan merupakan kasus non-linear (Santosa & Umam, 2018).

Rectified Linear Unit (ReLU) merupakan salah satu fungsi aktivasi yang sering digunakan pada *Convolutional Neural Network* (Chen, Sun, & Wang, 2018). Bentuk fungsi ReLU:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

Pada fungsi aktivasi ReLU, semua nilai x negatif akan dipetakan ke 0, seperti pada (2).



Sumber: *tinymind.com*

Gambar 7 Grafik fungsi ReLU

Kelebihan fungsi ReLU yaitu:

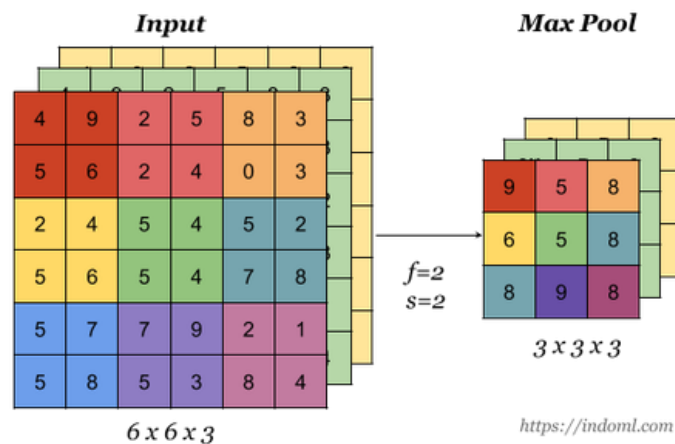
1. Fungsi ReLU konvergen terhadap *stochastic gradient descent* dibandingkan dengan fungsi sigmoid/tanh.
2. Operasi neuron pada fungsi ReLU lebih ringan dibanding fungsi *sigmoid/tanh* yang melibatkan operasi eksponensial. ReLU hanya melakukan *threshholding* sebuah matriks aktivasi pada nilai 0. (Chen, Sun, & Wang, 2018)

2.1.3.3 Pooling

Tujuan utama dari *pooling* yaitu untuk mengurangi kompleksitas pada lapisan selanjutnya. Dalam hal citra, *pooling* akan mengurangi resolusi citra. Hal ini akan mengurangi jumlah parameter yang harus diperbarui sehingga mempercepat komputasi dan mengatasi *overfitting*. *Pooling* tidak berpengaruh terhadap jumlah lapisan.



Salah satu jenis *pooling* yang umum digunakan yaitu *max pooling* (Albawi, Mohammed, & Al-Zawi, 2017).



Sumber: indoml.com

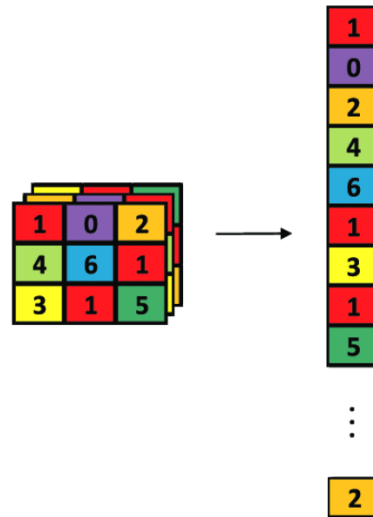
Gambar 8 Ilustrasi operasi *pooling*

Pada operasi *pooling*, ada 2 argumen yang perlu diperhatikan, yaitu *filter* dan *stride*. *Filter* merupakan ukuran matriks yang digunakan untuk melakukan *pooling*. Adapun *stride* merupakan banyaknya perpindahan kolom dan/atau baris setelah melakukan satu operasi *pooling*. Misalnya pada gambar 8, *filter* yang digunakan berukuran 2x2 dengan *stride* sebanyak 2. Operasi yang digunakan yaitu *max pooling*. Operasi dilakukan dari kiri atas hingga kanan bawah. Hasil dari setiap operasi *max pooling* merupakan nilai maksimal dari lingkup *filter*nya. Pada operasi pertama, pada lingkup *filter*nya terdapat nilai 4, 9, 5, dan 6. Hasil dari operasi tersebut yaitu 9, karena merupakan nilai tertinggi diantara keempat nilai yang lain (Santosa & Umam, 2018).

2.1.3.4 Flatten

Flatten merupakan metode untuk mengubah data matriks n dimensi menjadi 1 dimensi. *Flatten* ini digunakan setelah *output* proses akhir dari ekstraksi fitur. *Output* dalam bentuk matriks n dimensi akan diubah menjadi 1 dimensi untuk dilakukan klasifikasi pada *full connected layer*.



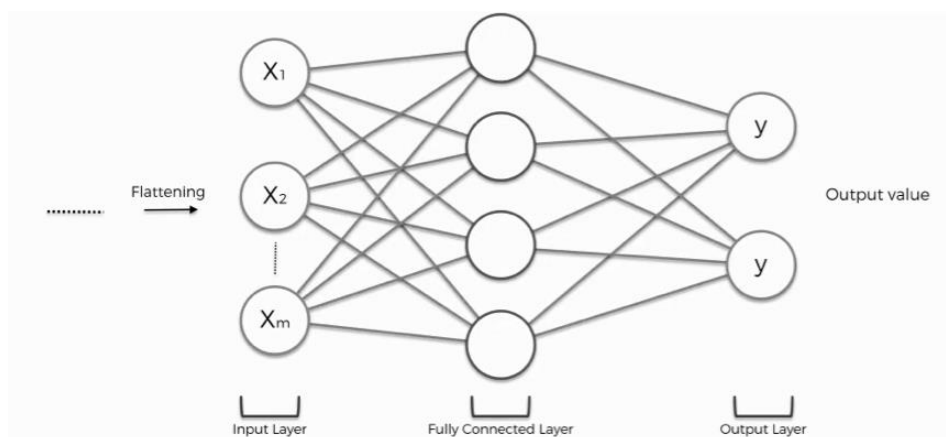


Sumber: researchgate.net

Gambar 9 Ilustrasi *flattening*

2.1.3.5 *Fully Connected Layer*

Fully connected layer merupakan bagian pada CNN yang mirip dengan *neural network* biasa. Setiap node dalam lapisan ini saling terhubung langsung dengan setiap node di lapisan sebelum dan berikutnya. *Fully connected layer* ini merupakan lapisan yang paling banyak menggunakan parameter dan membutuhkan waktu komputasi yang tinggi saat *training* (Albawi, Mohammed, & Al-Zawi, 2017). Pada gambar 10, input layer pada FC merupakan hasil flattening seperti pada gambar 9.

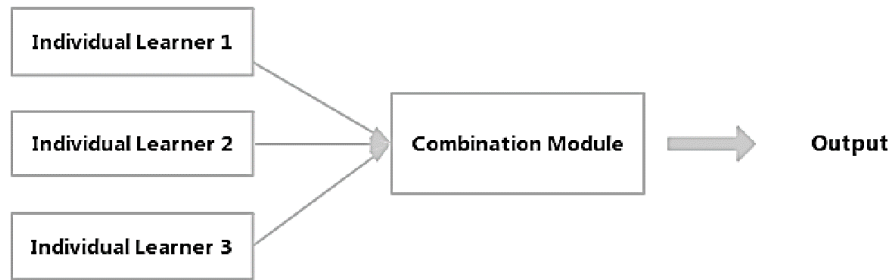


Sumber: superdatascience.com

Gambar 10 *Fully connected layer*



2.1.4 Ensemble Learning



Sumber: (Lee, Chen, Yu, & Lai, 2017)

Gambar 11 Struktur *ensemble learning*

Ensemble learning merupakan metode dalam *machine learning* yang digunakan untuk mencari solusi terbaik. *Ensemble* bekerja dengan cara melakukan *training* pada beberapa jenis metode atau model kemudian mengombinasikannya. *Ensemble learning* biasanya akan memberikan hasil yang lebih baik dibanding jika hanya satu model. Penelitian tentang *ensemble* banyak digunakan pada model dengan hasil yang kurang bagus untuk meningkatkan akurasi. *Ensemble learning* memiliki beberapa kelebihan. Pertama, secara statistik, *ensemble learning* dapat mengurangi kemungkinan kesalahan hipotesis dibandingkan jika hanya satu model. Kedua, *ensemble learning* dapat mengurangi risiko *overfitting* (Lee, Chen, Yu, & Lai, 2017).

Salah satu metode *ensemble* yaitu *boosting*. Metode *boosting* merupakan metode yang dapat meningkatkan ketelitian dalam proses klasifikasi dan prediksi dengan cara membangkitkan kombinasi dari suatu model. Namun, hasil klasifikasi atau prediksi yang dipilih merupakan model yang memiliki bobot yang paling besar. *Adaptive Boosting (AdaBoost)* merupakan salah satu algoritma *ensemble learning* terbaik. *AdaBoost* dipopulerkan oleh Freund dan Schapire pada 1999. *AdaBoost* akan memadukan model yang lemah dan akan mengonversi menjadi model yang baik pada akhirnya. *Adaboost* melatih pengklasifikasian secara sekuensial pada setiap iterasi. Pengklasifikasian dasar dilatih dengan menggunakan data latih dengan koefisien bobot yang bergantung dari performa pengklasifikasi

sebelumnya untuk memberikan bobot yang lebih besar pada data yang diklasifikasi. Jika pengklasifikasi telah dilatih sebanyak yang dikehendaki,



maka seluruh pengklasifikasian akan dikombinasikan untuk membentuk suatu keputusan akhir terhadap model yang menunjukkan performa terbaik.

Berikut merupakan cara pembobotan pada algoritma AdaBoost (Yang, Chen, Yan, Zhao, & Fan, 2017):

1. Input data $S = \{x_i, y_i\}$, $i = 1, 2, 3, \dots, N$, $y_i \in Y, Y = c_1, c_2, \dots, c_k$, c_k adalah label, dan jumlah model T .

2. Inisialisasi bobot sampel pelatihan

$$w_i = \frac{1}{N}, i = 1, 2, \dots, N \quad (3)$$

3. Untuk $t = 1, \dots, T$

a. Menghitung kesalahan klasifikasi

$$h_{tc} = X \rightarrow \{c_1, c_2, \dots, c_k\} \quad (4)$$

$$\varepsilon_{tc} = \sum_{i=1}^N w_i (y_i \neq h_{tc}(x_i)) \quad (5)$$

b. Menghitung indeks kepercayaan dihitung:

$$\alpha_{tc} = \frac{1}{2} \left(\frac{1 - \varepsilon_{tc}}{\varepsilon_{tc}} \right) \quad (6)$$

c. Perbaharui bobot sampel pelatihan

$$w_i = \left(\frac{w_i}{Z_t} \right) \begin{cases} \exp(-\alpha_{tc}) & \text{jika } y_i = h(x_i) \\ \exp(\alpha_{tc}) & \text{jika } y_i \neq h(x_i) \end{cases} \quad (7)$$

4. Output pembelajaran terakhir, kombinasi semua klasifikasi

$$H = \operatorname{argmax} \left(\sum_{t=1}^T s_t h_t \right) \quad (8)$$

Keterangan:

w_i = bobot

N = banyaknya data

ε_{tc} = error rate

x_i = data ke-i

y_i = kelas asli data ke-i

kelas prediksi

indeks kepercayaan model

total bobot



2.1.5 Evaluasi Model

Untuk melakukan evaluasi terhadap model yang telah dibuat, perlu diukur performa dari model tersebut. Secara umum, yang digunakan sebagai ukuran performa model klasifikasi adalah akurasi. Akurasi adalah jumlah prediksi benar untuk semua kelas dibagi dengan jumlah datanya. Namun, untuk mengukur performa lebih lanjut, ukuran yang sangat populer digunakan adalah dengan menggunakan *confusion matrix* (Santosa & Umam, 2018). *Confusion matrix* untuk kelas sebanyak n (Garillos-Manliguez, 2016):

		Prediksi			
		Kelas 1	Kelas 2	...	Kelas n
Aktual	Kelas 1	x_{11}	x_{12}	...	x_{1n}
	Kelas 2	x_{21}	x_{22}	...	x_{2n}

	Kelas n	x_{n1}	x_{n2}	...	x_{nn}

Untuk menghitung *Total True Negative* (TTN), *Total False Negative*(TFN), *Total False Positive* (TFP), *Total True Positive*(TTP) sebagai berikut:

$$TFN_i = \sum_{\substack{j=1 \\ j \neq i}}^n x_{ij} \quad (9)$$

$$TFP_i = \sum_{\substack{j=1 \\ j \neq i}}^n x_{ji} \quad (10)$$

$$TTN_i = \sum_{\substack{j=1 \\ j \neq i}}^n \sum_{\substack{k=1 \\ k \neq i}}^n x_{jk} \quad (11)$$

$$TTP_{all} = \sum_{j=1}^n x_{jj} \quad (12)$$

$$TP_i = x_{ii} \quad (13)$$

Untuk menghitung *Presisi*(P), *Recall* (R) dan *Specificity*(S):

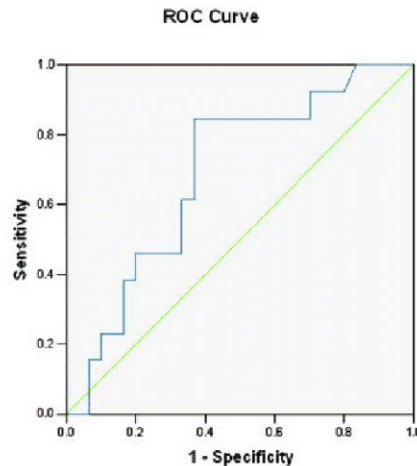
$$P_i = \frac{TP_i}{TP_i + TFP_i} \quad (14)$$

$$R_i = \frac{TP_i}{TP_i + TFN_i} \quad (15)$$



$$S_i = \frac{TTN_{all}}{TTN_{all} + TFP_i} \quad (16)$$

$$Akurasi = \frac{TTP_{all}}{Total\ data} \quad (17)$$



Gambar 12 Kurva ROC

Selain menggunakan *confusion matrix*, salah satu cara dalam mengukur performa model klasifikasi adalah menggunakan kurva ROC (*Receiver Operating Characteristic*). Kurva ROC ini menunjukkan *trade-off* (tawar menawar) antara *recall* dengan *false alarm rate (1-Specificity)*. *Recall* biasa disebut *True Positive Rate (TPR)* sedangkan *false alarm rate* biasa disebut *False Positive Rate (FPR)*. Model klasifikasi yang paling baik adalah model yang memiliki nilai TPR setinggi mungkin dan memiliki FPR yang sekecil mungkin. (Santosa & Umam, 2018).

2.2 Penelitian Terkait

2.2.1 Deteksi Penyakit Daun Tanaman Menggunakan Teknik *Soft Computing* dan Segmentasi Citra

Penyakit tanaman yang banyak dan sulit dikenali berakibat pada tingkat produktivitas pada bidang pertanian. Hal ini yang menjadi alasan utama Vijai Singh untuk melakukan penelitian ini pada 2017. Deteksi penyakit tanaman dilakukan

otomatis untuk melakukan pemantauan terhadap lahan yang luas merupakan salah besar. Vijai Singh menggunakan algoritma dengan teknik segmentasi citra untuk deteksi otomatis dan mengklasifikasikan penyakit pada daun tanaman. Citra daun diambil menggunakan kamera. Selanjutnya, kualitas citra



ditingkatkan dengan melakukan distorsi dan mengambil bagian penting saja. Bagian berwarna hijau kemudian ditandai kemudian dihapus. Tahap ini merupakan tahap segmentasi. Setelah itu, akan dilakukan ekstraksi fitur dan klasifikasi. Berdasarkan hasil percobaan, akurasi hasil klasifikasi mencapai 95,71% menggunakan *Support Vector Machine* (SVM) (Singh & Misra, 2017).

2.2.2 Klasifikasi Tanaman menggunakan *Convolutional Neural Network*

Penggunaan *machine learning* seperti *deep neural network* pada data pertanian mendapat perhatian besar dalam beberapa tahun terakhir. Salah satu masalah dalam bidang pertanian yaitu klasifikasi otomatis terhadap spesies tanaman berdasarkan tipenya. Klasifikasi tersebut berakibat pada pemberian pestisida, pemupukan, dan pemanenan pada spesies yang berbeda. Penelitian ini menggunakan CNN untuk mengklasifikasikan tipe tanaman menggunakan citra. Penelitian ini menghasilkan akurasi hingga 97,47% menggunakan CNN (Yalcin & Razavi, 2016).

2.2.3 Deteksi Penyakit Tanaman Tomat menggunakan Pemrosesan Citra

Sagar Petal melakukan penelitian untuk mendeteksi penyakit tanaman tomat. Untuk mendeteksi penyakitnya, dilakukan identifikasi terhadap daun tanaman tomat. Identifikasi dilakukan menggunakan pemrosesan terhadap citra daun tomat. Ada empat penyakit tanaman tomat yang diidentifikasi menggunakan *image segmentation* dan *multi class SVM*. Akurasi yang dihasilkan mencapai 93,75% dari model yang didapatkan. (Vetal & Khule, 2017)

2.2.4 Klasifikasi Penyakit Tanaman Tomat menggunakan Algoritma *Deep Learning*

Rangarajan dalam penelitiannya pada 2018 membuat metode deteksi penyakit tanaman tomat dengan cara yang cepat. Hal ini dilakukan untuk meningkatkan kualitas dan kuantitas produksi tomat. Penelitiannya menggunakan data citra daun tomat (6 jenis penyakit dan sehat) untuk mengidentifikasi penyakitnya. Data citra daun tomat diolah menggunakan dua arsitektur CNN, yaitu AlexNet dan VGG16Net. Sebanyak 13.262 citra diolah hingga menghasilkan akurasi 97,29%

VGG16Net dan 97,49% untuk AlexNet. (Rangarajan, Purushothaman, & 2018)



2.2.5 *Adaboost* dan CNN untuk Klasifikasi Kendaraan

Wei Chen dalam penelitiannya pada 2018 membuat model untuk mengklasifikasikan lima jenis kendaraan yang berbeda menggunakan algoritma *adaboost* dan *deep convolutional neural network* (CNN). Pada saat *train* data, dilakukan ekstraksi fitur menggunakan CNN. Hasil ekstraksi fiturnya akan diklasifikasi menggunakan beberapa model SVM. Proses *training* akan menghasilkan banyak model. Berbagai model yang dihasilkan akan dilakukan metode *ensemble* menggunakan algoritma *adaboost* untuk menghasilkan model baru. Hasil uji coba dari model yang dibuat menghasilkan akurasi yang tinggi mencapai 99,50% pada data *test*, serta hanya membutuhkan 28 ms untuk mengidentifikasi suatu citra kendaraan. Performa dari model ini secara signifikan lebih baik dibanding algoritma tradisional seperti SIGT-SVM, HOG-SVM dan SURF-SVM. Bahkan ekstraksi fitur menggunakan CNN dapat mengurangi parameter pembelajaran sebelum *train* menggunakan model SVM. Proses ini dapat menurunkan penggunaan penyimpanan dengan akurasi yang tetap tinggi. (Chen, Sun, & Wang, 2018)

2.2.6 Algoritma Klasifikasi Ensemble untuk Convolutional Neural Network berbasis *Adaboost*

AdaBoost merupakan salah satu algoritma *ensemble learning* yang dapat melakukan klasifikasi dengan baik. Shuo Yang mengombinasikan Algoritma *AdaBoost* dengan *Convolutional Neural Network* (CNN) yang diberi nama ACNN. Metode ACNN menggunakan metode ensemble yang berbeda dengan *AdaBoost* pada umumnya. ACNN menggunakan metode training baru, bobot bukan hanya diberikan pada masing-masing *base classifier*, tapi juga pada setiap kelas pada setiap *base classifier*. Uji coba dilakukan terhadap beberapa dataset seperti MNIST, ETH-80, dan USPS. Hasilnya, ACNN dapat mengurangi *error rate* klasifikasi dan mengatasi masalah kelas tidak seimbang (Yang, Chen, Yan, Zhao, & Fan, 2017).

