

SKRIPSI

**EVALUASI KINERJA PENERAPAN *MULTI-CHANNEL
TRANSFER LEARNING* CNN PADA KLASIFIKASI
CITRA PENYAKIT DAUN PADI**

Disusun dan diajukan oleh

KHAWARITZMI ABDALLAH AHMAD

H071171005



PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

**EVALUASI KINERJA PENERAPAN *MULTI-CHANNEL TRANSFER LEARNING* CNN PADA
KLASIFIKASI CITRA PENYAKIT DAUN PADI**

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**

KHAWARITZMI ABDALLAH AHMAD

H071171005

**PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN
MATEMATIKA**

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Khawaritzmi Abdallah Ahmad
NIM : H071171005
Program Studi : Sistem Informasi
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

EVALUASI KINERJA PENERAPAN *MULTI-CHANNEL TRANSFER LEARNING* CNN PADA KLASIFIKASI CITRA PENYAKIT DAUN PADI

adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 28 Juni 2021

Yang menyatakan,



Khawaritzmi Abdallah Ahmad

NIM: H071171005

EVALUASI KINERJA PENERAPAN *MULTI-CHANNEL TRANSFER LEARNING CNN* PADA KLASIFIKASI CITRA PENYAKIT DAUN PADI

Disusun dan diajukan oleh

KHAWARITZMI ABDALLAH AHMAD

H071171005

Telah diperhatikan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Juni 2021 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

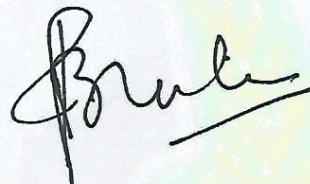
Pembimbing Utama,



Dr. Eng. Armin Lawi, S.Si., M.Eng

NIP: 19720423 199512 1 001

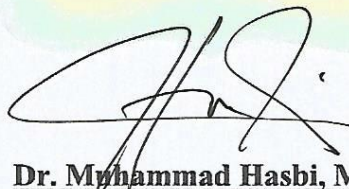
Pembimbing Pertama,



Sri Astuti Thamrin, S.Si., M.Stat., Ph.D

NIP: 19740713 199903 2 001

Ketua Program Studi,



Dr. Muhammad Hasbi, M.Sc.

NIP: 19630720 198903 1 003

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Khawaritzmi Abdallah Ahmad
NIM : H071171005
Program Studi : Sistem Informasi
Judul Skripsi : Evaluasi Kinerja Penerapan Multi-Channel Transfer Learning CNN pada Klasifikasi Citra Penyakit Daun Padi

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Tanda tangan

Ketua : Dr. Eng. Armin Lawi, S.Si., M.Eng
Sekretaris : Sri Astuti Thamrin, S.Si., M.Stat., Ph.D
Anggota : Dr. Muhammad Hasbi, M.Sc.
Anggota : Supri bin Hj. Amir, S.Si., M.Eng.

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

Ditetapkan di : Makassar
Tanggal : 28 Juni 2021



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Strata 1 yang berjudul “**Evaluasi Kinerja Penerapan Multi-Channel Transfer Learning CNN pada Klasifikasi Citra Penyakit Daun Padi**”. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini. Oleh karena itu, pada kesempatan ini dengan segala kerendahan hati penulis menyampaikan terima kasih yang setulus-tulusnya kepada :

1. Keluarga, ayahanda tercinta **Dr. Ahmad Talib M.Si** dan ibunda tersayang **Dra. Hartini Tahir M.Pd** yang telah memberikan dukungan baik moril maupun materiil serta doa yang tiada henti-hentinya kepada penulis. Serta ucapan terima kasih juga pada kakak saya **Intifadah Alijah Ahmad** dan **Ifititah Annisa Ahmad** serta adik saya **Vasha Jadyah Al fadillah Ahmad**.
2. Rektor Universitas Hasanuddin beserta jajarannya, Bapak Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam beserta jajarannya, dan seluruh pihak birokrasi atas pengetahuan yang diberikan, baik dalam bidang akademik maupun bidang kemahasiswaan
3. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng.** selaku pembimbing utama dan Ibu **Sri Astuti Thamrin, S.Si., M.Stat., Ph.D.** selaku pembimbing pertama untuk segala ilmu dan kesabaran dalam membimbing dan mengarahkan penulis, serta bersedia meluangkan waktunya untuk mendampingi penulis sehingga skripsi ini dapat terselesaikan. Serta kepada Bapak **Dr. Muhammad Hasbi, M.Sc.** selaku Ketua Program Studi Sistem Informasi sekaligus dan Bapak **Supri Bin Hj Amir, S.Si., M.Eng.** atas kesediannya menjadi anggota tim penguji yang telah bersedia meluangkan waktunya untuk memberikan saran dan arahan kepada penulis dalam penyusunan skripsi ini.
4. Dosen Departemen Matematika, dan terkhusus kepada ibu dan bapak dosen Program Studi Sistem Informasi Fakultas MIPA Universitas Hasanuddin untuk

semua ilmu yang telah diberikan kepada penulis selama menempuh pendidikan di jenjang strata

5. Pegawai dan staf Science Building terutama **Bapak Suardi** yang senantiasa membantu pengurusan dokumen.
6. Teman-teman seperjuangan **Program Studi Sistem Informasi 2017** yang telah mendukung dan berjuang bersama-sama dalam suka dan duka. atas waktu dan tenaga yang telah diberikan selama penulis menyelesaikan skripsinya, terutama **Muhammad Arizki** yang selalu membukan pintu ruang belajar, **Fadhil Hidayat Amin, Farhan Muhammad, Muhammad Muflihun Naim, Restu Adi Akbar, Kennedy, Aris Akhyar Abdillah, Nur Khairunisa, Eka Kurnia, Eka Fitriani, Nurfadila Firdani Salam, Fadhilah Putri Taha, Geby Nionsi, Siti Rabiatul Adawiyah, Azzahra Mubarikah, dan Muthia Amanah Arum** dan teman-teman asisten yang lain sebagai teman mengajar dan belajar pemrograman selama kuliah.
7. Teman-teman dan kakak-kakak di Team CEROS dan Team DarkSides terutama **Muhammad Rafli Nur Ihsan, Agung Almuqtadir dan Dwipa Furqan** yang ikut membantu lancarnya proses pengerjaan skripsi.
8. Kakak-kakak dan adik-adik **Program Studi Sistem Informasi 2014, 2015, 2016, 2018, 2019, dan 2020.**
9. Komunitas **Stack Overflow, Kaggle, dan Github** yang telah membantu penulis dan *programmer* lain di belahan dunia lainnya untuk menemukan jalan keluar dari setiap permasalahan dalam menyusun kode program.
10. Semua pihak yang telah membantu penulis dan tak sempat penulis tuliskan satu persatu

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Makassar, 28 Juni 2021

Penulis

**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK
KEPENTINGAN AKADEMIS**

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

Nama : Khawaritzmi Abdallah Ahmad
NIM : H071171005
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

**Evaluasi Kinerja Penerapan Multi-Channel Transfer Learning CNN pada
Klasifikasi Citra Penyakit Daun Padi**

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 28 Juni 2021

Yang menyatakan



(Khawaritzmi Abdallah Ahmad)

ABSTRAK

Padi adalah tanaman penghasil beras yang menjadi sumber bahan pokok di Indonesia. Produksi padi di Indonesia sepanjang Januari hingga September 2020 mengalami penurunan dibandingkan produksi di 2019. Identifikasi penyakit daun padi adalah salah satu solusi utama meningkatkan produktivitas. Salah satu metode yang dapat digunakan untuk melakukan identifikasi adalah dengan melakukan metode klasifikasi menggunakan salah satu kelas *deep feedforward artificial neural network* yang banyak diaplikasikan pada analisis citra yaitu *Convolutional Neural Networks* (CNN). Untuk beberapa kasus, jumlah data yang tersedia sedikit dan terbatas. Akibatnya model yang dihasilkan memiliki kinerja jauh di bawah ekspektasi. Oleh karena itu penelitian ini menggunakan metode *multi-channel* model berbasis dua arus CNN dan tiga arus CNN menggunakan *transfer learning* dengan mengkombinasikan *ResNet50*, *InceptionV3*, *Xception*, dan *VGG16* sebagai *base model*. Hasilnya implementasi arsitektur *multi-channel transfer learning* CNN menghasilkan kinerja dengan akurasi training 100% dan akurasi testing 95.83%. Nilai precision, recall, dan f1-score pada model yang dihasilkan juga diatas 0.85 hingga 1.0, dan untuk nilai AUC pada kurva ROC tiap-tiap kelas data citra penyakit padi yang digunakan mencapai 0.95 hingga 1.0. Secara umum seluruh kinerja yang dihasilkan oleh model dari arsitektur *multichannel transfer learning* CNN lebih baik dibandingkan model yang dihasilkan oleh arsitektur transfer learning sederhana.

Kata Kunci: *Multichannel CNN*, Penyakit Daun Padi, *Transfer Learning*.

ABSTRACT

Rice is a rice-producing plant which is a source of staple food in Indonesia. The production of rice in Indonesia during January to September 2020 decreased compared to the production in 2019. Identification of rice leaf disease is one of the main solutions to increase the productivity. One method to identify is performing a classification method using one of the deep feedforward artificial neural network classes that are widely applied to image analysis, namely Convolutional Neural Networks (CNN). In some cases, the amount of data available is small and limited. As a result, the resulting model has a performance which is far from expectations. Therefore, this study uses a multi-channel model based on two and three CNN streams using transfer learning by combining ResNet50, InceptionV3, Xception, and VGG16 as the base model. The result is that the implementation of CNN's multi-channel transfer learning architecture produces performance with 100% training accuracy and 95.83% testing accuracy. The precision, recall, and f1-score values in the resulting model are also above 0.85 to 1.0, and the AUC value on the ROC curve for each class of rice disease image data used reaches 0.95 to 1.0. In general, the overall performance generated by the model from the CNN multichannel transfer learning architecture is better than the model generated by the simple transfer learning architecture.

Keywords: *Multichannel CNN, Rice Leaf Disease, Transfer Learning*

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERNYATAAN KEASLIAN	ii
HALAMAN PERSETUJUAN PEMBIMBING	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR	vii
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan masalah	3
1.5 Manfaat Penelitian	3
1.6 Organisasi Skripsi	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. <i>Convolutional Neural Network</i>	5
2.1.1. Konvolusi	6
2.1.2. <i>Pooling</i>	7

2.1.3.	<i>Flatten</i>	7
2.1.4.	<i>Dropout</i>	8
2.1.5.	<i>Rectifier Linear Unit (ReLU)</i>	9
2.1.6.	<i>Fully Connected Layer</i>	10
2.2.	Penyakit Daun Padi	10
2.2.1.	<i>Bacterial Leaf Blight</i>	10
2.2.2.	<i>Brown Spot</i>	11
2.2.3.	<i>Leaf Smut</i>	12
2.3.	Model Warna Citra Digital	13
2.3.1.	<i>Grayscale</i>	13
2.3.2.	<i>RGB</i>	14
2.3.3.	<i>HSV</i>	15
2.4.	Arsitektur <i>Convolutional Neural Network</i>	15
2.4.1.	<i>VGG16</i>	15
2.4.2.	<i>ResNet50</i>	16
2.4.3.	<i>InceptionV3</i>	16
2.4.4.	<i>Xception</i>	17
2.5.	<i>Transfer Learning</i>	18
2.6.	<i>Multi-Channel CNN</i>	20
2.7.	Augmentasi Data	22
2.8.	Evaluasi Kinerja Model	23
2.8.1.	<i>Confusion Matrix</i>	23
2.8.2.	Presisi	24
2.8.3.	<i>Recall</i>	24

2.8.4.	Akurasi dan Validasi Akurasi	24
2.8.5.	<i>F1-Score</i>	25
2.8.6.	<i>Loss Function</i>	25
2.8.7.	Kurva AUC-ROC.....	26
BAB 3 METODE PENELITIAN.....		27
3.1.	Waktu dan Lokasi Penelitian	27
3.2.	<i>Dataset</i>	27
3.3.	Instrumen Penelitian.....	27
3.4.	Tahap Penelitian.....	27
3.5.	<i>Pre-processing</i>	28
3.6.	<i>Split Data</i>	29
3.7.	<i>Training CNN</i>	29
3.7.1.	<i>Transfer Learning</i> dan <i>Fine Tuning</i>	29
3.7.2.	Implementasi <i>Multi-Channel CNN</i>	30
3.8.	Evaluasi.....	31
BAB IV PEMBAHASAN.....		32
4.1.	Deskripsi Data.....	32
4.2.	<i>Pre-processing</i>	32
4.3.	<i>Split Data</i>	34
4.4.	Augmentasi Data.....	34
4.5.	Arsitektur <i>Convolutional Neural Network</i>	35
4.6.	<i>Transfer Learning</i> Menggunakan Model Warna RGB	36
4.6.1.	Akurasi	36
4.6.2.	<i>Confusion Matrix</i>	38

4.6.3. Kurva ROC	40
4.7. <i>Transfer Learning</i> Menggunakan Model Warna <i>Grayscale</i>	41
4.7.1. Akurasi	41
4.7.2. <i>Confusion Matrix</i>	43
4.7.3. Kurva ROC	44
4.8. <i>Transfer Learning</i> Menggunakan Model Warna HSV	46
4.8.1. Akurasi	46
4.8.2. <i>Confusion Matrix</i>	48
4.8.3. Kurva ROC	49
4.9. Multichannel <i>Transfer Learning</i> CNN	50
4.9.1. Akurasi	52
4.9.2. <i>Confusion Matrix</i>	54
4.9.3. Kurva ROC	54
4.10. Evaluasi Kinerja.....	55
BAB V KESIMPULAN DAN SARAN.....	59
5.1. Kesimpulan	59
5.2. Saran.....	59
DAFTAR PUSTAKA	60
LAMPIRAN	64

DAFTAR TABEL

Tabel 2. 1. Confusion matrix	24
Tabel 4. 1. Hasil evaluasi kinerja dihasilkan model menggunakan citra model warna RGB	39
Tabel 4. 2. Hasil evaluasi kinerja dihasilkan model menggunakan citra model warna Grayscale	44
Tabel 4. 3. Hasil evaluasi kinerja dihasilkan model menggunakan citra model warna HSV	49
Tabel 4. 4. Urutan kinerja model dari yang terbaik pada masing-masing model transfer learning	51
Tabel 4. 5. Hasil evaluasi kinerja dihasilkan model menggunakan arsitektur Multi-channel transfer learning CNN	54
Tabel 4. 6. Tabel Evaluasi precision, recall, dan f1-score.	55
Tabel 4. 7. Tabel Evaluasi Akurasi dan Loss	57
Tabel 4. 8. Tabel AUC-ROC.....	57

DAFTAR GAMBAR

Gambar 2. 1. Arsitektur Convolutional Neural Network.....	5
Gambar 2. 2. Ilustrasi Konvolusi	6
Gambar 2. 3. Ilustrasi Proses Max Pooling dan Average Pooling	7
Gambar 2. 4. Ilustrasi proses Flattening dari hasil pooling pada Feature Map.....	8
Gambar 2. 5. Ilustrasi proses training yang berbeda antara (a) jaringan syaraf biasa dan (b) jaringan syaraf dengan metode dropout.	9
Gambar 2. 6. Kurva fungsi aktivasi ReLU.....	9
Gambar 2. 7. Ilustrasi Penerapan Fully connected.....	10
Gambar 2. 8. Daun padi yang terkena penyakit Bacterial Leaf Blight	11
Gambar 2. 9. Daun padi yang terkena penyakit Brown Spot.....	12
Gambar 2. 10. Daun padi yang terkena penyakit Leaf Smut	12
Gambar 2. 11. Citra Digital Grayscale.....	14
Gambar 2. 12. Citra Digital RGB	14
Gambar 2. 13. Arsitektur VGG16.....	16
Gambar 2. 14. Arsitektur ResNet50.....	16
Gambar 2. 15. Arsitektur InceptionV3.....	17
Gambar 2. 16. Arsitektur Xception	18
Gambar 2. 17. Ilustrasi proses learning yang berbeda antara (a) machine learning tradisional dan (b) transfer learning.	19
Gambar 2. 18. Arsitektur Multi-channel Convolutional Neural Network.	21
Gambar 2. 19. Contoh Penerapan Augmentasi Data.....	23
Gambar 2. 20. Kurva AUC - ROC	26
Gambar 3. 1. Diagram Alur Penelitian.....	28
Gambar 3. 2. Cara kerja Transfer Learning	30
Gambar 3. 3. Implementasi setiap dataset pada pretrained network yang digunakan	30

Gambar 4. 1. Jumlah citra pada masing-masing kelas	32
Gambar 4. 2. Citra setelah diduplikasi dari RGB menjadi format HSV dan Grayscale	33
Gambar 4. 3. Hasil resize citra menjadi ukuran 299×299 piksel	33
Gambar 4. 4. Ilustrasi proses rescale sebelum dilakukan proses rescale (kiri). Setelah dilakukan rescale (kanan)	34
Gambar 4. 5. Ilustrasi hasil proses augmentasi pada satu citra	35
Gambar 4. 6. Arsitektur transfer learning sederhana menggunakan Xception(a), InceptionV3(b), ResNet50(c), dan VGG16(d) sebagai pretrained network	36
Gambar 4. 7. (kiri) Kurva akurasi pada setiap epoch pada data latih dan data validasi. (kanan) Kurva loss pada setiap epoch pada data latih dan data validasi, menggunakan citra pada model warna RGB	38
Gambar 4. 8. Confusion Matrix yang dihasilkan model menggunakan model warna RGB	39
Gambar 4. 9. Kurva AUC-ROC yang dihasilkan model menggunakan citra model warna RGB	41
Gambar 4. 10. (kiri) Kurva akurasi pada setiap epoch pada data latih dan data validasi. (kanan) Kurva loss pada setiap epoch pada data latih dan data validasi, menggunakan citra pada model warna Grayscale	43
Gambar 4. 11. Confusion Matrix yang dihasilkan model menggunakan citra model warna Grayscale	44
Gambar 4. 12. Kurva AUC-ROC yang dihasilkan model menggunakan citra model warna Grayscale	45
Gambar 4. 13. (kiri) Kurva akurasi pada setiap epoch pada data latih dan data validasi. (kanan) Kurva loss pada setiap epoch pada data latih dan data validasi, menggunakan citra pada model warna HSV	47
Gambar 4. 14. Confusion Matrix yang dihasilkan model menggunakan citra model warna HSV	48

Gambar 4. 15. Kurva AUC-ROC yang dihasilkan model menggunakan citra model warna HSV	50
Gambar 4. 16. Arsitektur yang dihasilkan arsitektur Multi-channel transfer learning CNN	52
Gambar 4. 17. (kiri) Kurva akurasi pada setiap epoch pada data latih dan data validasi. (kanan) Kurva loss pada setiap epoch pada data latih dan data validasi, menggunakan citra pada arsitektur Multi-channel transfer learning CNN	53
Gambar 4. 18. Confusion Matrix yang dihasilkan model menggunakan arsitektur Multi-cahnnel transfer learning CNN	54
Gambar 4. 19. Kurva AUC-ROC yang dihasilkan model menggunakan arsitektur Multi-channel transfer learning CNN	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Padi adalah tanaman penghasil beras yang menjadi sumber bahan pokok di Indonesia. Beras saat ini masih belum tergantikan oleh bahan makanan pokok yang lain seperti jagung, kentang, ubi, dan sagu. Realisasi panen padi sepanjang Januari hingga September 2020 sebesar 9,01 juta hektar, atau mengalami penurunan sekitar 275,35 ribu hektar (2,97 persen) dibandingkan 2019 yang sebesar 9,28 juta hektar. Produksi padi di Indonesia sepanjang Januari hingga September 2020 diperkirakan sekitar 45,45 juta ton Gabah Kering Giling (GKG), atau mengalami penurunan sekitar 1,49 juta ton (3,17 persen) dibandingkan produksi di 2019 yang sebesar 46,94 juta ton GKG(BPS, 2020).

Banyak hal yang menyebabkan produksi padi lambat dan kurang produktif. Salah satu penyebab utamanya adalah penyakit padi (Ramesh & Vydeki, 2020). Identifikasi penyakit daun padi adalah salah satu solusi utama meningkatkan produktivitas. Hal ini juga dapat membantu dalam pemilihan pestisida dan mencegah penyebaran lebih lanjut (Premi dkk., 2019). Biasanya deteksi penyakit daun padi dilakukan oleh para ahli hanya dengan mata telanjang yang dapat menghabiskan waktu yang lama dan rentan terhadap kesalahan pada saat melakukan identifikasi penyakit daun padi (Ramesh & Vydeki, 2020).

Pada tahun 2019, Kementerian ATR/BPN menetapkan luas lahan baku sawah nasional 2019 berdasarkan Keputusan Menteri ATR/Kepala BPN No. 686/SK-PG.03.03/XII/2019, tanggal 17 Desember 2019, tentang Penetapan Luas Lahan Baku Sawah Nasional Tahun 2019 yaitu sebesar 7.463.948 hektar (BPS, 2020). Karena luas lahan sawah di Indonesia yang mencapai 7.463.948 hektar, diharapkan penyakit daun padi dapat diidentifikasi secara cepat, akurat, dan tidak menghabiskan waktu yang lama. Metode yang dapat digunakan untuk melakukan identifikasi penyakit daun padi dengan cepat yaitu membuat sebuah model untuk melakukan pendeteksian penyakit daun padi dengan media citra menggunakan metode klasifikasi.

Salah satu metode yang dapat digunakan untuk melakukan klasifikasi adalah dengan menggunakan *Artificial Neural Network* (Suyanto, 2018). *Artificial Neural Network* (ANN) adalah sistem pemrosesan komputasi yang terinspirasi oleh cara sistem saraf biologis (seperti otak manusia) beroperasi (O'Shea & Nash, 2015). Kelas *deep feedforward artificial neural network* yang banyak diaplikasikan pada analisis citra adalah *Convolutional Neural Networks* atau biasa disingkat CNN (Suyanto, 2018).

Untuk mengimplementasikan CNN terhadap metode klasifikasi, diperlukan data citra yang digunakan sebagai *input*. Untuk beberapa kasus, jumlah data yang tersedia sedikit dan terbatas. Akibatnya model yang dihasilkan memiliki kinerja jauh di bawah ekspektasi dan juga memberikan akurasi yang rendah sehingga model tersebut tidak dapat dikatakan model yang baik (Barman dkk., 2019). *Transfer learning* merupakan teknik *deep learning* dimana model dilatih untuk belajar dan menyimpan pengetahuan dari satu masalah dan menggunakan model yang sama untuk masalah lain yang sejenis. Selanjutnya model CNN yang sudah dilatih dari kumpulan data besar diimplementasikan pada klasifikasi citra yang lain (Rajayogi dkk., 2019). Untuk itu mengatasi masalah dataset yang terbatas dapat digunakan metode *transfer learning*.

Saat menggunakan *transfer learning*, file model telah dilatih dengan sejumlah besar citra dari dataset *computer vision* (ImageNet), dan kemudian dioptimalkan (*fine-tuned*) menggunakan citra penyakit daun padi. Selanjutnya dilakukan pengembangan menggunakan *multi-channel* model berbasis dua arus CNN (*double channel*) dan tiga arus CNN (*three channel*) menggunakan *transfer learning* yang ada (Miyra dkk., 2020). Oleh karena itu dengan menggunakan metode ini akan didapatkan model untuk klasifikasi penyakit daun padi dengan kinerja yang baik walaupun dengan dataset citra yang terbatas.

Aplikasi metode *Multi-Channel Transfer Learning Convolutional Neural Network* untuk melakukan klasifikasi data citra penyakit daun padi masih belum banyak diteliti, Oleh karena itu masalah ini penting diteliti agar dapat memberikan pengetahuan mengenai bagaimana kinerja dari metode *Multi-Channel Transfer Learning CNN* yang diaplikasikan pada data citra penyakit daun padi.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, dapat dirumuskan masalah sebagai berikut:

1. Bagaimanakah kinerja model *Multi-Channel transfer learning CNN* apabila diaplikasikan pada data citra penyakit daun padi dengan jumlah yang terbatas?
2. Apakah kinerja arsitektur *Multi-channel transfer learning CNN* memberikan hasil klasifikasi data citra penyakit daun padi yang lebih baik, apabila dibandingkan dengan arsitektur *Transfer learning CNN* ?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menganalisis kinerja yang dihasilkan oleh arsitektur *Multi-Channel Transfer learning Convolutional neural network* untuk klasifikasi data citra penyakit daun padi dengan data citra yang terbatas.
2. Membandingkan kinerja arsitektur *Multi-channel transfer learning CNN* dan *Transfer learning CNN* untuk klasifikasi data citra penyakit daun padi.

1.4 Batasan masalah

Batasan masalah pada penelitian ini adalah:

1. Menggunakan data citra daun padi.
2. Arsitektur yang digunakan pada *pretrained network* untuk Metode *Transfer learning CNN* adalah *ResNet50*, *InceptionV3*, *Xception*, dan *VGG16*.
3. Penyakit daun padi yang diteliti adalah *Brown spot*, *Leaf smut*, and *Bacterial blight*.
4. Arsitektur *Multi-Channel CNN* menggunakan *Double Channel CNN* dan *Three-Channel CNN*.

1.5 Manfaat Penelitian

Penelitian ini dapat menghasilkan model *machine learning* yang nantinya dapat digunakan pada alat pendeteksi penyakit daun padi. Selain itu, hasil dari penelitian ini juga dapat memberikan informasi tentang implementasi metode *Multi-channel transfer learning CNN* untuk klasifikasi data citra dengan data citra yang terbatas.

1.6 Organisasi Skripsi

Bab I Pendahuluan pertama-tama menjelaskan latar belakang permasalahan, solusi dan metode yang digunakan beserta hasil yang diharapkan. Berdasarkan latar belakang tersebut, disusun rumusan masalah, batasan masalah, tujuan penelitian dan manfaat penelitian. Pada akhir Bab ini dijelaskan organisasi skripsi untuk memudahkan pembaca mengikuti arah penelitian ini.

Bab II Tinjauan Pustaka secara garis besarnya menjelaskan tentang konsep pengolahan citra digital, metode klasifikasi CNN beserta arsitektur yang digunakan dan pengembangan arsitektur CNN. Penjelasan konsep-konsep ukuran dalam mengevaluasi kinerja metode CNN yang berbasis pada tabel *confusion matrix*, akurasi, presisi, *recall*, *loss*, *F1-Score*, dan Kurva ROC.

Bab III Metode Penelitian menjelaskan secara detail tentang bagaimana peneliti melakukan penelitian berdasarkan alur penelitian yang dibuat mulai dari *preprocessing data*, *Split data*, *Training data* menggunakan arsitektur *Multi-Channel Transfer learning CNN*, Evaluasi kinerja model yang dihasilkan, Waktu dan Lokasi Penelitian, *Dataset* yang digunakan, serta Instrumen Penelitian.

Bab IV Hasil dan Pembahasan, menjelaskan hasil-hasil penelitian dalam hal ini evaluasi kinerja beberapa arsitektur yang digunakan pada *pretrained network* dari arsitektur *Multi-Channel Transfer learning CNN* dibandingkan dengan *Transfer learning CNN* sederhana untuk klasifikasi citra penyakit daun padi. Pada akhir Bab diberikan pembahasan sebagai tesis dari hasil perbandingan,

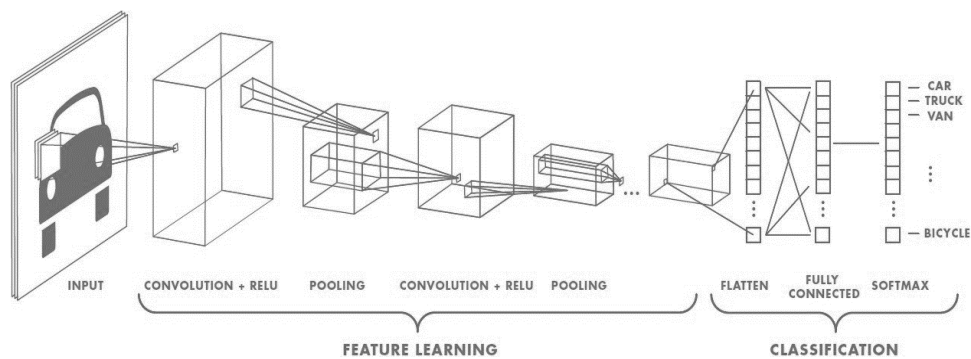
Bab V Kesimpulan dan Saran menjelaskan tentang kesimpulan yang menjawab tujuan dari penelitian dan saran untuk kelanjutan penelitian.

BAB II

TINJAUAN PUSTAKA

2.1. Convolutional Neural Network

Convolutional Neural Network (CNN), juga disebut *ConvNet*, adalah jenis *Artificial Neural Network* (ANN) atau dalam bahasa Indonesia disebut Jaringan Syaraf Tiruan (JST). ANN memiliki arsitektur umpan-maju yang dalam dan memiliki kemampuan generalisasi yang luar biasa dibandingkan dengan jaringan lain dengan lapisan *fully connected*, CNN dapat mempelajari fitur-fitur dari objek terutama data spasial dan dapat mengidentifikasi objek dengan lebih efisien. Model *deep CNN* umumnya terdiri dari dua tahap, yang pertama adalah *feature extraction* atau *feature learning* dan kedua adalah *classification* atau klasifikasi dengan *fully connected layer*. Model konseptual dasar CNN ditunjukkan pada Gambar 2.1 (Ghosh dkk., 2019).



Gambar 2. 1. Arsitektur Convolutional Neural Network

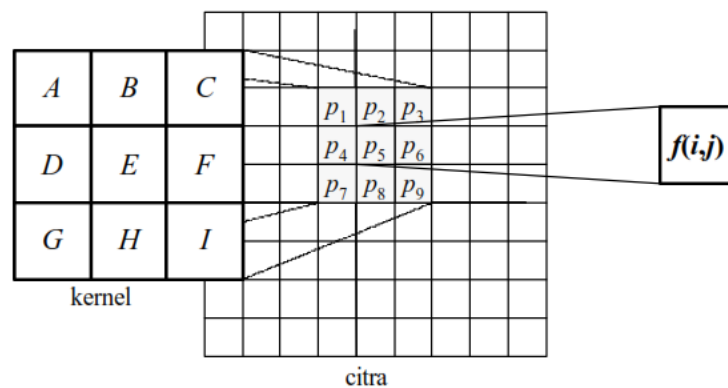
Perbedaan antara CNN dan JST/ANN tradisional adalah CNN digunakan terutama di bidang pengenalan pola dalam citra sehingga dapat mengurangi jumlah parameter saat membuat model (O'Shea & Nash, 2015). Asumsi terpenting tentang masalah yang diselesaikan oleh CNN seharusnya tidak memiliki fitur yang bergantung secara spasial. Dengan kata lain, misalnya dalam sebuah aplikasi deteksi wajah, tidak perlu memperhatikan letak wajah di dalam citra (Albawi dkk., 2017). Berikut ini dijelaskan tahapan CNN.

2.1.1. Konvolusi

Konvolusi merupakan salah satu tahap yang sangat penting pada arsitektur CNN (Suyanto, 2018). Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah *kernel/filters* pada seluruh bagian citra sehingga menghasilkan keluaran berupa fitur map (Eka Putra, 2016). Pada Gambar 2.2 mengilustrasikan bagaimana operasi konvolusi menggunakan *kernel* berukuran 3×3 pada *input* citra 9×9 . Persamaan untuk konvolusi ditunjukkan pada Persamaan (2.1).

$$S(i, j) = (I * K)(i, j) = \sum_a \sum_b I(a, b)K(i - a, j - b) \tag{2.1}$$

Hasil operasi konvolusi dengan *kernel* 3×3 akan berkurang satu piksel untuk masing-masing sisi dari ukuran inputnya. Jika digunakan *kernel* ukuran 5×5 maka akan berkurang sebanyak 2 piksel pada masing-masing sisi, dan seterusnya untuk ukuran-ukuran lain. Agar didapatkan hasil dengan ukuran yang sama dengan *input*, salah satu cara yang dapat digunakan adalah dengan menggunakan *zero padding* pada *input*. Untuk kasus *kernel* 3×3 maka dapat ditambahkan masing-masing satu piksel pada setiap sisi *input*, dengan nilai piksel pada *padding* ini adalah nol (Santosa & Umam, 2018).

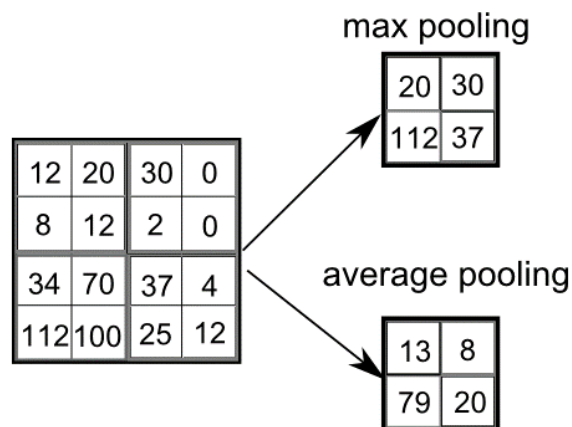


Gambar 2. 2. Ilustrasi Konvolusi

2.1.2. Pooling

Pooling layer berfungsi menjaga ukuran data setelah proses *convolution* dengan melakukan *down sampling* (pereduksian sampel), yaitu mengambil *feature map* dengan ukuran lebih besar dan menyusutkannya ke *feature map* berukuran lebih rendah. Dengan *pooling* data direpresentasikan kedalam bentuk yang lebih kecil, sehingga dapat mereduksi waktu komputasi dan mengatasi *overfitting* (Suyanto, 2018).

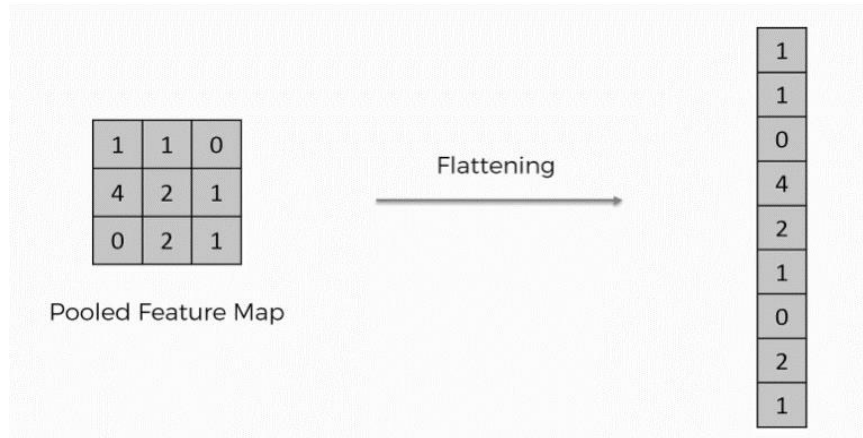
Ada berbagai jenis teknik yang digunakan pada *pooling layers* seperti *max pooling*, *min pooling*, *average pooling*, *gated pooling*, *tree pooling*, dan lain-lain. *Max pooling* adalah teknik penyatuan yang paling populer dan paling banyak digunakan. *Max pooling* bekerja dengan mempartisikan citra ke sub-wilayah persegi, dan hanya mengembalikan nilai maksimum dari dalam sub-wilayah itu seperti pada Gambar 2.3. Salah satu ukuran filter paling umum yang digunakan dalam *max-pooling* adalah 2x2 (Gosh dkk, 2020).



Gambar 2. 3. Ilustrasi Proses *Max Pooling* dan *Average Pooling*

2.1.3. Flatten

Flatten merupakan metode untuk mengubah data matriks hasil dari *Pooling Layers* 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 *fully connected layer*.

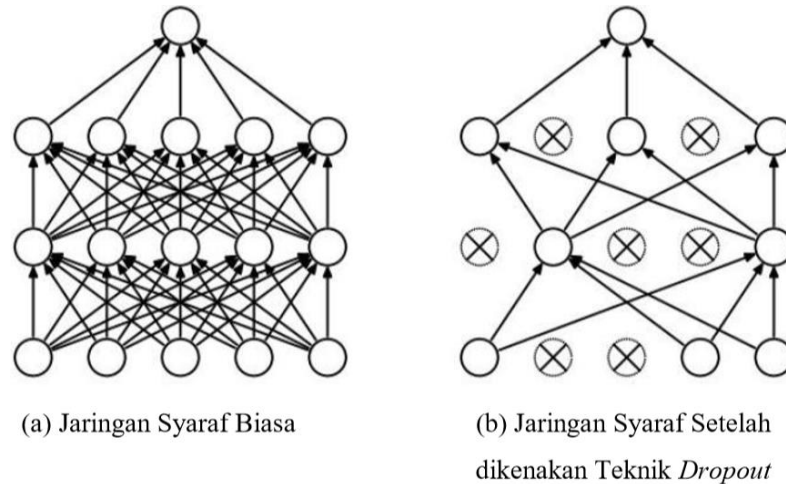


Gambar 2. 4. Ilustrasi proses *Flattening* dari hasil *pooling* pada *Feature Map*

2.1.4. Dropout

Dropout adalah salah satu pendekatan yang paling banyak digunakan untuk regularisasi. Metode ini secara acak melakukan *drop neuron* pada jaringan selama proses *training*. Dengan melakukan *drop* pada beberapa neuron, kemampuan dari *feature selection* (seleksi fitur) dapat di distribusikan ke semua *neuron* secara merata dan secara langsung memaksa model untuk mempelajari beberapa fitur independen.

Dengan melakukan *drop* pada *neuron*, berarti *neuron* yang *didrop* tidak akan mengambil bagian dalam propagasi maju atau propagasi mundur selama proses *training*. Tetapi dalam kasus proses pengujian, jaringan skala penuh digunakan untuk melakukan prediksi. Pada Gambar 2.5 ditunjukkan gambaran cara kerja *dropout* pada arsitektur CNN saat proses *training* berlangsung (Gosh dkk, 2020). *Dropout* merupakan salah satu usaha untuk mencegah terjadinya *overfitting* dan juga mempercepat proses *learning* (Santoso & Ariyanto, 2018).

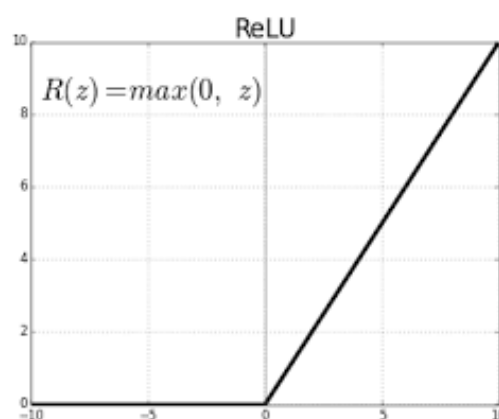


Gambar 2. 5. Ilustrasi proses *training* yang berbeda antara (a) jaringan syaraf biasa dan (b) jaringan syaraf dengan metode *dropout*.

2.1.5. *Rectifier Linear Unit (ReLU)*

Rectifier Linear Unit (ReLU) adalah fungsi aktivasi yang paling umum digunakan di jaringan *neural network*. ReLU digunakan untuk mengubah semua nilai input menjadi bilangan positif. Keuntungan dari ReLU adalah membutuhkan beban komputasi yang sangat minimal dibandingkan dengan yang lain (Ghosh dkk., 2019). Representasi matematis dari ReLU terlihat pada Persamaan 2.2.

$$ReLU(x) = \max \{0, x\} \quad (2.2)$$

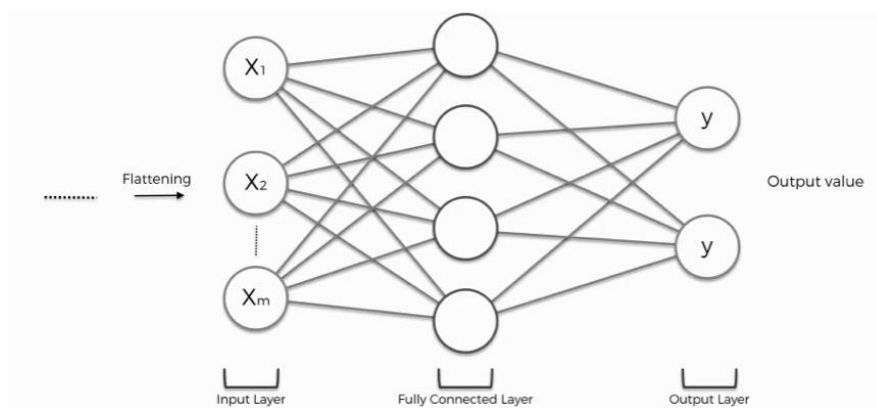


Gambar 2. 6. Kurva fungsi aktivasi ReLU

2.1.6. Fully Connected Layer

Fully Connected Layer atau *Dense Layer* merupakan *layer* atau lapisan paling akhir yang digunakan beberapa arsitektur CNN, di mana setiap *neuron* di dalam *fully connected layer* terhubung dengan setiap *neuron* dari *layer* sebelumnya. Lapisan terakhir dari *fully connected layer* digunakan sebagai lapisan keluaran (*classifier*) dari arsitektur CNN. *Fully connected layer* adalah jenis jaringan saraf tiruan umpan-maju (*Artificial Neural Network*) dan mengikuti *multi-layer perceptron neural network* tradisional (MLP).

Fully connected layer mengambil masukan dari lapisan konvolusional atau *pooling*, yang berupa kumpulan metrik (peta fitur) dan metrik tersebut diratakan (dengan metode *Flatten*) untuk membuat vektor dan vektor ini kemudian dimasukkan ke dalam *fully connected layer* untuk menghasilkan hasil akhir (Gosh dkk, 2020).



Gambar 2. 7. Ilustrasi Penerapan *Fully connected*

2.2. Penyakit Daun Padi

Tanaman padi dipengaruhi oleh berbagai jenis penyakit. Mereka dapat dikelompokkan dalam varietas yang berbeda seperti penyakit yang dipengaruhi oleh bakteri, penyakit yang disebabkan oleh virus, penyakit dan gangguan jamur (Premi dkk., 2019). Fokus penelitian ini adalah penyakit berikut ini.

2.2.1. Bacterial Leaf Blight

Penyakit disebabkan oleh bakteri *Xanthomonas oryzae* pv. *oryzae* (*Xoo*). Patogen ini dapat menginfeksi tanaman padi pada semua fase pertumbuhan

tanaman dari mulai persemaian sampai menjelang panen. Penyebab penyakit (patogen) menginfeksi tanaman padi pada bagian daun melalui luka daun atau lobang alami berupa stomata dan merusak klorofil daun. Hal tersebut menyebabkan menurunnya kemampuan tanaman untuk melakukan fotosintesis yang apabila terjadi pada tanaman muda mengakibatkan mati dan pada tanaman fase generatif mengakibatkan pengisian gabah menjadi kurang sempurna (Balai Besar Penelitian Tanaman Padi, 2015).

- Bagian tanaman yang terpengaruh umumnya mengenai daun tanaman.
- Bentuk gejalanya berupa luka memanjang di ujung daun, luka berukuran beberapa inci dengan warna luka kuning ke putih karena pengaruh bakteri.



Gambar 2. 8. Daun padi yang terkena penyakit *Bacterial Leaf Blight*
(archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases)

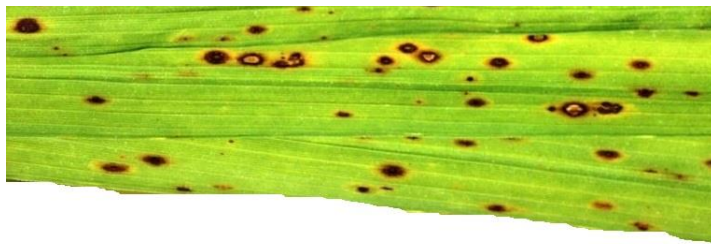
2.2.2. *Brown Spot*

Hubungan antara terjadinya penyakit *Brown Spot* dengan ketersediaan unsur hara tanah sangat erat. Tanaman yang kurang sehat sangat mudah terserang penyakit ini. Pada kondisi tanah yang kekurangan terhadap zat unsur kalium penyakit bercak coklat dapat menimbulkan kerugian hasil 50 sampai 90 persen. Faktor lain yang berpengaruh adalah sistem drainase yang tidak baik, sehingga mengganggu terserapnya unsur-unsur hara (Herlisa, 2019).

Gejala yang paling umum dari penyakit ini adalah adanya bercak berwarna coklat tua, berbentuk oval sampai bulat, berukuran sebesar biji wijen, pada permukaan daun, pada pelepah atau pada gabah. Gejala khas penyakit ini adalah adanya bercak coklat pada daun berbentuk oval yang merata di permukaan daun dengan titik tengah berwarna abu-abu atau putih. Titik abu-abu di tengah bercak

merupakan gejala khas penyakit bercak daun coklat pada permukaan daun. Bercak yang masih muda berwarna coklat gelap atau keunguan berbentuk bulat. Pada varietas yang peka panjang bercak dapat mencapai panjang 1 cm (Herlisa, 2019).

- Bagian tanaman yang terpengaruh umumnya mengenai daun tanaman.
- Bentuk gejala penyakit adalah bulat hingga lonjong dengan warna luka coklat kemerahan sampai coklat tua.



Gambar 2. 9. Daun padi yang terkena penyakit *Brown Spot*
(archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases)

2.2.3. *Leaf Smut*

Leaf Smut adalah penyakit jamur ringan dimana bintik-bintik hitam kecil timbul terutama di daun. Bintik atau pustula yang menonjol pecah dan melepaskan spora yang terbawa udara. Infeksi seringkali cukup berat untuk mematikan ujung daun (Texas Plant Disease Handbook, n.d. 2021).

- Bagian tanaman yang terpengaruh umumnya mengenai daun tanaman.
- Bentuk gejala penyakit berupa bintik-bintik kecil berserakan di seluruh daun dalam bentuk yang tidak seragam dengan warna luka coklat kemerahan.



Gambar 2. 10. Daun padi yang terkena penyakit *Leaf Smut*
(archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases)

2.3. Model Warna Citra Digital

Citra digital adalah fungsi dua dimensi $f(x, y)$ yang merupakan proyeksi dari pemandangan 3-dimensi menjadi bidang proyeksi 2-dimensi, dimana x, y merepresentasikan lokasi elemen citra atau piksel dan mengandung nilai intensitas cahaya. Jika nilai x, y dan intensitas diskrit, maka citra tersebut dikatakan citra digital. Secara matematis, citra digital adalah representasi matriks dari citra dua dimensi menggunakan sejumlah elemen sel titik yang terbatas, yang biasanya disebut piksel (elemen citra, atau pel) (Persamaan 2.3) (Tyagi, 2018).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \cdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix} \quad (2.3)$$

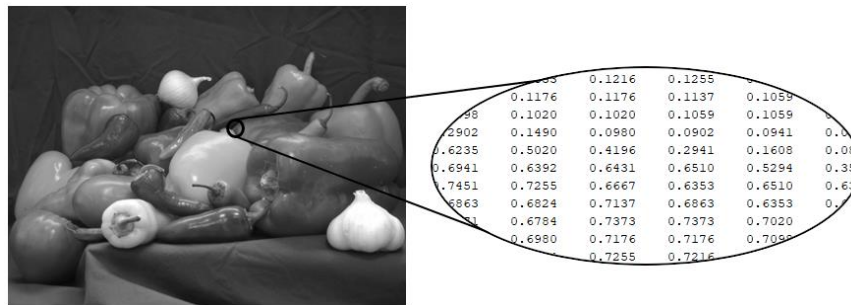
Setiap piksel diwakili oleh nilai numerik. Untuk citra skala abu-abu, satu nilai yang mewakili intensitas piksel (biasanya dalam rentang $[0, 255]$) sudah cukup. Untuk citra berwarna, tiga nilai (mewakili jumlah merah (R), hijau (G), dan biru (B)) disimpan. Jika suatu citra hanya memiliki dua intensitas, maka citra tersebut disebut citra biner (Tyagi, 2018). Jenis-jenis citra yang digunakan dijelaskan berikut ini.

2.3.1. *Grayscale*

Citra *grayscale* merupakan citra yang nilai intensitas pikselnya didasarkan pada derajat keabuan. Pada citra *grayscale* 8-bit, derajat warna hitam sampai dengan putih dibagi ke dalam 256 derajat keabuan di mana warna putih sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0 (Pamungkas, 2017). Rentang warna pada hitam dan putih sangat cocok digunakan untuk pengolahan file citra. Salah satu bentuk fungsinya digunakan dalam kedokteran (X-ray). Hitam dan putih sebenarnya merupakan hasil rata-rata dari citra berwarna. Dengan demikian persamaannya dapat dituliskan pada Persamaan (2.4) (RD.Kusumanto, 2011).

$$I_{bw}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \tag{2.4}$$

dimana $I_R(x, y)$ = nilai piksel merah titik (x, y) , $I_G(x, y)$ = nilai piksel hijau titik (x, y) , $I_B(x, y)$ = nilai piksel biru titik (x, y) , sedangkan $I_{BW}(x, y)$ = nilai piksel hitam dan putih atau abu-abu titik (x, y) .



Gambar 2. 11. Citra Digital *Grayscale*

2.3.2. RGB

Citra berwarna atau RGB (*Red, Green, Blue*). Pada citra berwarna ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (*Red*), hijau (*Green*) dan biru (*Blue*). Jika masing-masing warna memiliki *range* 0-255, maka totalnya adalah $255^3 = 16.581.375$ (16 K) variasi warna berbeda pada citra, dimana variasi warna ini cukup untuk citra apapun. Karena jumlah bit yang diperlukan untuk setiap piksel, citra tersebut juga disebut citra-bit warna. Citra berwarna ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau dan biru untuk setiap pikselnya, seperti yang ditunjukkan Gambar 2.12 (RD. Kusumanto, 2011).



Gambar 2. 12. Citra Digital *RGB*

2.3.3. HSV

Selain model RGB terdapat juga model HSV dimana model ini terdapat 3 komponen yaitu, hue, saturation, dan value. Hue adalah suatu ukuran panjang gelombang yang terdapat pada warna dominan yang diterima oleh penglihatan sedangkan, Saturation adalah ukuran banyaknya cahaya putih yang bercampur pada hue (Kusumanto dkk., 2011).

2.4. Arsitektur *Convolutional Neural Network*

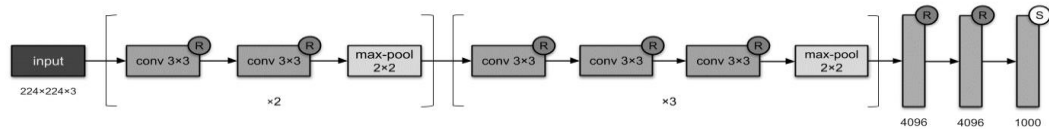
Arsitektur CNN merupakan gabungan dari beberapa *layer* yang secara umum terdiri dari *Convolutional Layer*, *Subsampling Layer (Pooling Layer)*, dan *Fully Connected Layer* (O'Shea & Nash, 2015). layer ini tersusun tanpa adanya aturan yang universal dan berbeda-beda tergantung dari dataset yang digunakan. Ada banyak arsitektur CNN yang telah didesain para ahli, yang pertama adalah LeNet (LeCun dkk., 1998) yang digunakan untuk membaca kode pos dan digit (Suyanto, 2018), namun karena keterbatasan perangkat dan waktu komputasi yang tinggi maka arsitektur ini sempat dilupakan karena dianggap tidak efektif saat itu.

Selanjutnya arsitektur CNN terus berkembang dan pada tahun 2012 muncullah *AlexNet* yang membuat arsitektur CNN kembali dilirik untuk aplikasi *Computer Vision* (Suyanto, 2018). Dari beberapa arsitektur yang didesain beberapa mengikuti kompetisi *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* yang menggunakan dataset *ImageNet* untuk melihat kinerja dari setiap arsitektur. Arsitektur dengan performa terbaik pada *ILSVRC* dapat digunakan untuk menjadi *base model* atau *pre-trained network* karena akurasi yang tinggi dan *validation loss* yang rendah. Beberapa arsitektur yang digunakan pada penelitian ini antara lain *InceptionV3*, *ResNet50*, *VGG16*, dan *Xception*.

2.4.1. *VGG16*

VGGNet merupakan arsitektur yang menunjukkan bahwa kedalaman jaringan merupakan komponen penting untuk menghasilkan performa yang tinggi. Salah satu jaringan *VGGNet* yang digunakan pada penelitian ini adalah *VGG16* yang mana terdapat 13 jaringan konvolusional dan 3 jaringan *fully connected*. *VGGNet* menggunakan filter konvolusi 3x3 dan *max-pooling* 2x2 diikuti oleh tiga lapisan

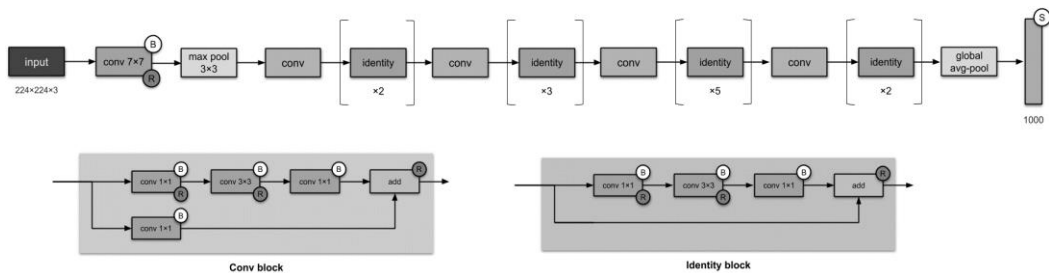
Tersambung Penuh (*fully connected*), dua yang pertama memiliki masing-masing 4096 saluran, yang ketiga berfungsi Klasifikasi *ILSVRC* 1000 arah dan karenanya berisi 1000 saluran (satu untuk setiap kelas) (Simonyan & Zisserman, 2015). *VGGNet* memiliki 138-140 juta parameter sehingga membutuhkan banyak memori.



Gambar 2. 13. Arsitektur *VGG16*

2.4.2. *ResNet50*

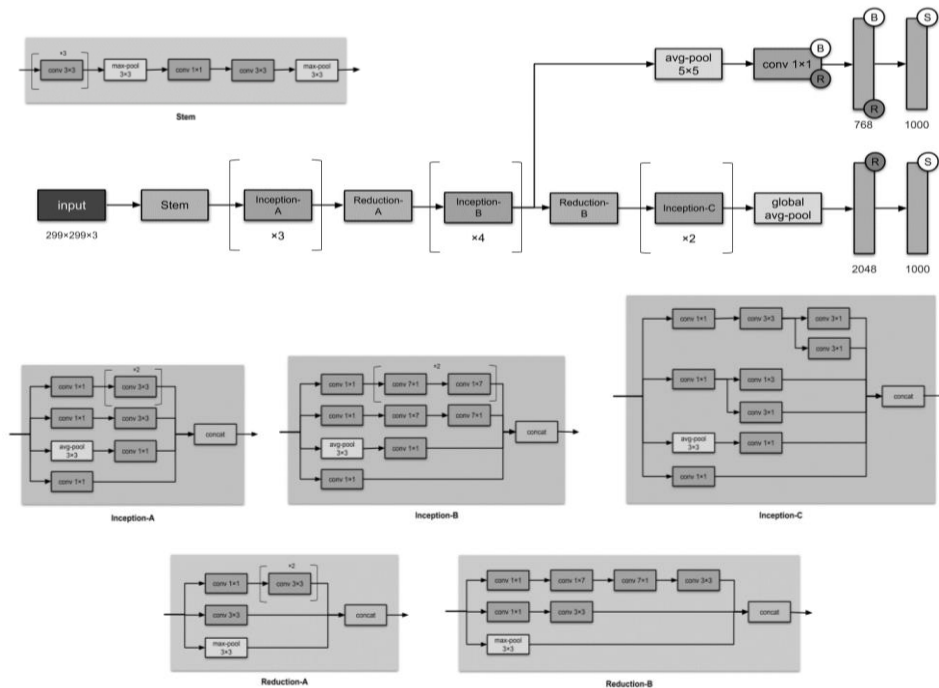
ResNet atau *residual network* memiliki perbedaan dari beberapa arsitektur yang berfokus pada kedalaman, *ResNet* mempunyai ciri khusus yaitu menggunakan koneksi pintasan (*shortcut connections*) dan tidak menggunakan *fully connected layers* di bagian akhir (He dkk., 2016).



Gambar 2. 14. Arsitektur *ResNet50*

2.4.3. *InceptionV3*

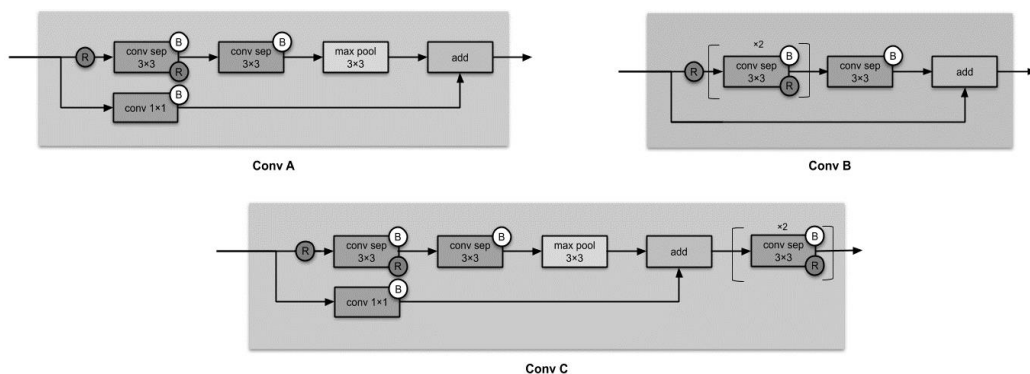
Inception-v3 adalah model pengenalan citra yang banyak digunakan yang memiliki ukuran *input* citra 299 x 299. Model yang dihasilkan arsitektur ini telah terbukti mencapai akurasi lebih dari 78,1% pada dataset *ImageNet*. Arsitektur ini adalah arsitektur jaringan *neural* konvolusional dari keluarga *Inception* yang membuat beberapa peningkatan dari versi sebelumnya termasuk menggunakan *label Smoothing*, konvolusi Difaktorkan 7 x 7, dan penggunaan *auxiliary classifier* untuk menyebarkan informasi label ke bagian bawah jaringan (bersama dengan penggunaan *batch normalization*) (Szegedy dkk., 2016).

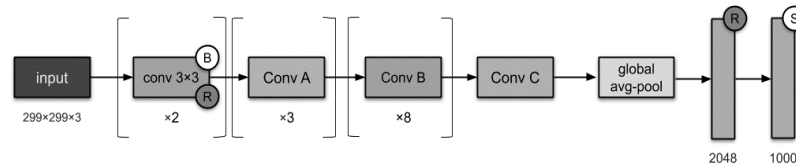


Gambar 2. 15. Arsitektur *InceptionV3*

2.4.4. *Xception*

Xception adalah adaptasi dari *Inception* yang dimana modul *Inception* telah diganti dengan konvolusi yang dapat dipisahkan secara mendalam. Singkatnya, arsitektur *Xception* adalah tumpukan linier dari lapisan konvolusi terpisah yang dapat dipisahkan dengan koneksi residual. Hal ini membuat arsitektur sangat mudah untuk didefinisikan dan dimodifikasi. Nama *Xception* merupakan singkatan dari "*Extreme Inception*" dan juga memiliki *input* citra yang sama dengan arsitektur *Inception* yaitu 299 x 299. (Chollet, 2017).

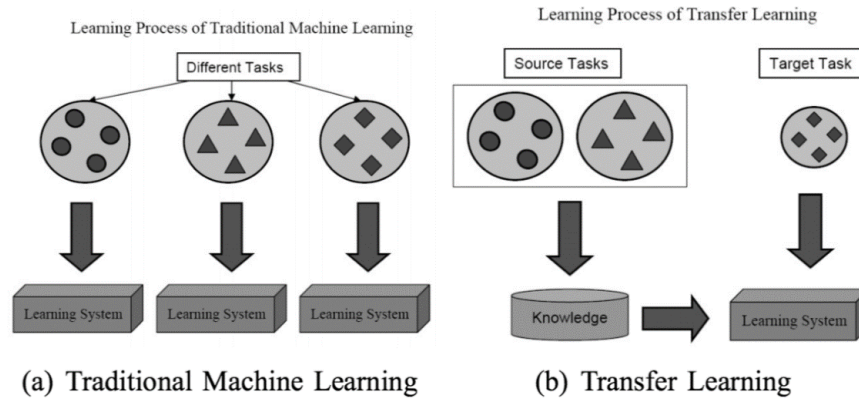


Gambar 2. 16. Arsitektur *Xception*

2.5. Transfer Learning

Transfer learning adalah metode menggunakan atau memanfaatkan kembali model atau pengetahuan yang telah dipelajari sebelumnya untuk peningkatan pembelajaran untuk tugas yang lebih baru dan berkaitan (Bali & Ghosh, 2018). Selain memberikan kemampuan untuk menggunakan kembali model yang sudah dibangun, *transfer learning* dapat membantu mempelajari tugas target dengan cara berikut:

- Peningkatan kinerja dasar: dengan menambah pengetahuan dari *isolated learner* (juga dikenal *ignored learner*) dengan pengetahuan dari model sumber atau model yang telah dilatih, kinerja dasar mungkin akan meningkat dengan menerapkan *transfer learning*.
- Waktu pengembangan model: Memanfaatkan pengetahuan dari model sumber mungkin juga membantu dalam mempelajari tugas target secara penuh, dibandingkan dengan model target yang belajar dari awal. Hal ini menghasilkan peningkatan dalam keseluruhan waktu yang dibutuhkan untuk mengembangkan / mempelajari model.
- Peningkatan kinerja akhir: Kinerja akhir yang lebih tinggi dapat dicapai dengan memanfaatkan *transfer learning*.



Gambar 2. 17. Ilustrasi proses *learning* yang berbeda antara (a) *machine learning* tradisional dan (b) *transfer learning*.

Secara formal *transfer learning* didefinisikan berikut ini. Sebuah domain D , didefinisikan sebagai dua komponen yang terdiri dari ruang fitur \mathcal{X} , dan probabilitas marginal $P(X)$, di mana X adalah titik data sampel. Di sini, $X = \{x_1, x_2, \dots, x_n\}$ dengan $x_i \in \mathcal{X}$. Jadi domain tersebut direpresentasikan secara matematis sebagai Persamaan (2.5) (Pan & Yang, 2010).

$$D = \{\mathcal{X}, P(X)\} \tag{2.5}$$

Sebuah tugas T , di sisi lain, dapat didefinisikan sebagai dua komponen yang terdiri dari ruang label Y , dan fungsi objektif η . Fungsi objektif juga dapat dilambangkan sebagai $P(Y|X)$ dari sudut pandang probabilistik menjadi Persamaan (2.6) (Pan & Yang, 2010).

$$T = \{Y, P(Y|X)\} = \{Y, \eta\} \tag{2.6}$$

Diberikan *source domain* D_s dan *learning task* T_s , *target domain* D_t dan *target task* T_t , *transfer learning* ditujukan untuk meningkatkan kemampuan pembelajaran dari *predictive function* atau *target task* pada D_t menggunakan *knowledge*/pengetahuan dari D_s dan T_s , dengan kondisi $D_s \neq D_t$ atau $T_s \neq T_t$ (Pan & Yang, 2010).

Skenario umum yang melibatkan *transfer learning* berdasarkan definisi sebelumnya adalah:

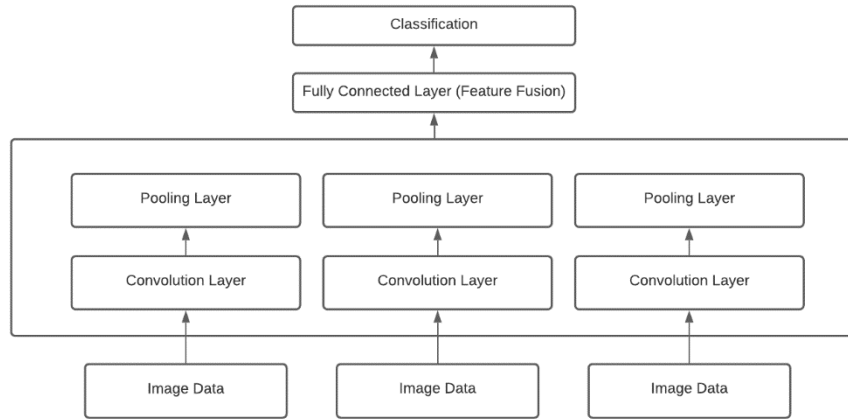
- a. $X_s \neq X_t$. Ruang fitur dari domain sumber dan target berbeda, misalnya dokumen tersebut ditulis dalam dua bahasa yang berbeda. Dalam konteks pemrosesan bahasa alami, hal ini umumnya disebut sebagai adaptasi lintas bahasa.
- b. $P(X_s) \neq P(X_t)$. Distribusi probabilitas marjinal dari domain sumber dan target berbeda, misalnya dokumen membahas topik yang berbeda. Skenario ini umumnya dikenal sebagai adaptasi domain.
- c. $Y_s \neq Y_t$. Ruang label antara dua tugas berbeda, misalnya dokumen perlu diberi label yang berbeda dalam tugas target. Dalam praktiknya, skenario ini biasanya terjadi dengan skenario 4, karena sangat jarang dua tugas berbeda memiliki ruang label yang berbeda, tetapi distribusi probabilitas bersyarat yang persis sama.
- d. $P(Y_s | X_s) \neq P(Y_t | X_t)$. Distribusi probabilitas bersyarat dari tugas sumber dan target berbeda, misalnya dokumen sumber dan target tidak seimbang sehubungan dengan kelas mereka.

2.6. Multi-Channel CNN

Arsitektur *Multi-Channel CNN* yang digunakan, menggunakan dua *channel* (*double-channel*) dan tiga *channel* (*three channel*) *input* model *convolutional network* yang merupakan penyempurnaan dari model CNN tradisional yang saat ini telah diaplikasikan pada perbandingan citra, seperti analisis sidik jari, analisis citra medis, pengenalan wajah, dan lain-lain (Jiang dkk., 2019). Jaringan dua cabang memungkinkan *patch1* dan *patch2* untuk mengekstrak vektor fitur melalui dua jaringan, selanjutnya *loss function* diterapkan pada dua vektor fitur di lapisan terakhir dan *network training* digabungkan, sehingga meningkatkan presisi (Jiang dkk., 2019).

Multi-channel CNN dalam hal ini *double-channel CNN* mewakili sebuah perbaikan berdasarkan jaringan dua cabang. Fitur-fitur digabungkan dalam *fully connected layer* pada CNN, yang merupakan pendekatan yang berbeda dari perbandingan fitur dari Jaringan dua cabang. CNN saluran ganda digunakan untuk target *recognition* dan klasifikasi citra dengan intensitas berbeda. *double-channel CNN*, dua jaringan saraf berada sepenuhnya independen. Bobot *convolution layer*

dan *pooling layer* juga independen. Dalam *fully connected layer*, fitur digabungkan dan *loss function* klasifikasi diturunkan berdasarkan fitur gabungan citra RGB dan citra *Gray Scale* untuk dua arus dan ditambahkan model warna HSV untuk tiga arus *multi-channel CNN*. Alur prosesnya terlihat seperti pada Gambar 2.18 (Jiang dkk., 2019).



Gambar 2. 18. Arsitektur *Multi-channel Convolutional Neural Network*.

Lapisan yang sepenuhnya terhubung mengubah semua peta fitur dua dimensi (2D) menjadi *input* untuk jaringan satu dimensi (1D) yang terhubung sepenuhnya. Saat memasukkan peta fitur 2D terakhir ke dalam jaringan 1D, metode yang paling mudah adalah menggabungkan semua peta fitur keluaran ke dalam vektor *input* panjang, seperti yang ditunjukkan pada Persamaan (2.7).

$$\begin{cases} u^l \rightarrow [U_1, U_2, U_3, \dots, U_i] \\ v^l \rightarrow [V_1, V_2, V_3, \dots, V_i] \\ w^l \rightarrow [W_1, W, W_3, \dots, W_i] \end{cases} \quad (2.7)$$

Penggabungan $[U_1, U_2, U_3, \dots, U_i]$, $[V_1, V_2, V_3, \dots, V_i]$, dan $[W_1, W, W_3, \dots, W_i]$ adalah dinyatakan sebagai Persamaan (2.8):

$$(X, Y) \rightarrow \begin{bmatrix} U_1, V_1, W_1 \\ U_2, V_2, W_2 \\ U_3, V_3, W_3 \\ \dots, \dots, \dots \\ U_i, V_i, W_i \end{bmatrix}$$

Atau

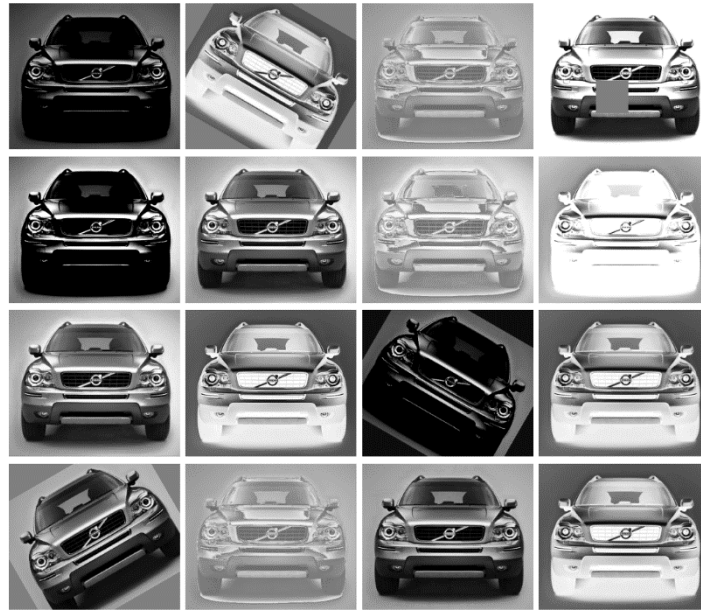
$$(X, Y) \rightarrow [U_1, U_2, U_3, \dots, U_i, V_1, V_2, V_3, \dots, V_i, W_1, W_2, W_3, \dots, W_i]^T \quad (2.8)$$

Berdasarkan *fully connected layer*, hasil akhir dihitung dari vektor jumlah kelas dari data (*num_classes_sized*) sebagai $[Y_1, Y_2, Y_3, \dots, Y_t]$. Selanjutnya dilakukan klasifikasi *softmax* berdasarkan keluaran dan hasil prediksi seperti yang ditunjukkan pada Persamaan (2.9).

$$\text{softmax}(Y_t) = \frac{\exp(Y_t)}{\sum \exp(Y_t)} \quad (2.9)$$

2.7. Augmentasi Data

Augmentasi Data adalah teknik yang digunakan untuk menambah atau memperluas ukuran *training dataset* secara artifisial. Pada proses augmentasi diterapkan operasi yang berbeda pada sampel data (hanya pada *training dataset*) dan mengubahnya secara artifisial menjadi satu atau banyak sampel data baru (versi baru), yang kemudian digunakan dalam proses pelatihan atau *training process*. Augmentasi Data penting karena terkadang terdapat kumpulan data latih berukuran sangat terbatas yang tersedia untuk sebagian besar masalah kompleks kehidupan nyata (misalnya, kumpulan data medis) dan fakta sebenarnya adalah bahwa semakin banyak sampel data latih dapat menghasilkan model CNN yang lebih baik (Ghosh dkk., 2019).



Gambar 2. 19. Contoh Penerapan Augmentasi Data

Ada beberapa operasi augmentasi data yang tersedia seperti *cropping*, *rotations*, *flipping*, *translations*, *contrast adjustment*, *scaling*, dan lain-lain seperti yang ditunjukkan pada Gambar 2.19. operasi tersebut dapat diterapkan secara terpisah atau dalam kombinasi untuk membuat beberapa versi baru dari satu sampel data. Alasan lain menggunakan metode ini adalah augmentasi data juga dapat memperkuat regularisasi dalam model CNN dengan menghindari masalah *overfitting* (Ghosh dkk., 2019).

2.8. Evaluasi Kinerja Model

2.8.1. *Confusion Matrix*

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining atau Sistem Pendukung Keputusan. Pada pengukuran kinerja menggunakan *confusion matrix* pada Tabel 2.1, T_x Menunjukkan jumlah prediksi data uji yang benar atau sama dengan kelas aktual dan $F_{x|y}$ Menunjukkan jumlah dari data uji yang diprediksikan salah atau berbeda dengan kelas aktual dari data uji.

Tabel 2. 1. *Confusion matrix*

ACTUAL Class	PREDICTED Class			
	Classes	Leaf Smut	Brown Spot	Bacterial Leaf Blight
Leaf Smut		T_{LS}	$F_{BS LS}$	$F_{BB LS}$
Brown Spot		$F_{LS BS}$	T_{BS}	$F_{BB BS}$
Bactrial Leaf Blight		$F_{LS BB}$	$F_{BS BB}$	T_{BB}

2.8.2. Presisi

Presisi (precision) merupakan rasio prediksi benar positif (TP) dibandingkan dengan keseluruhan hasil yang diprediksi positif. Berdasarkan *confusion matrix* yang digunakan rumus dari presisi pada penelitian ini dinyatakan pada Persamaan (2.10).

$$Precision_x = \frac{T_x}{T_x + \sum_y F_{y|x}} \quad (2.10)$$

2.8.3. Recall

Recall atau rasio TP adalah ukuran untuk berapa banyak TP yang diprediksi dari semua positif dalam kumpulan data, kadang juga disebut kepekaan (*sensitivity*). Rumus *recall* pada penelitian ini dapat dituliskan seperti pada Persamaan (2.11).

$$Recall_x = \frac{T_x}{T_x + \sum_y F_{x|y}} = \frac{T_x}{N_x} \quad (2.11)$$

2.8.4. Akurasi dan Validasi Akurasi

Akurasi digunakan untuk mengukur kinerja algoritma dengan cara yang dapat ditafsirkan. Akurasi suatu model biasanya ditentukan dalam bentuk persentase. Akurasi adalah ukuran seberapa akurat prediksi model dibandingkan dengan data

sebenarnya dan Akurasi dihitung berdasarkan data latih. Rumus untuk akurasi pada penelitian ini ditunjukkan pada Persamaan (2.12).

$$Accuracy = \frac{\sum_x T_x}{\sum_x T_x + \sum_x \sum_y F_{x|y}} = \frac{\sum_x T_x}{N} \quad (2.12)$$

Sedangkan validasi akurasi dihitung berdasarkan data validasi. Yang terbaik adalah mengandalkan validasi akurasi dari kinerja model, karena *neural network* yang baik pada akhirnya akan menyesuaikan data latih pada 100%, tetapi akan berkinerja buruk pada data yang baru ditemuinya.

2.8.5. *F1-Score*

F1-Score atau Skor F1 adalah skor *F* yang paling umum digunakan. Ini adalah kombinasi presisi dan *recall*, yaitu rata-rata harmoniknya. *F1-Score* dapat dihitung melalui Persamaan (2.13).

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (2.13)$$

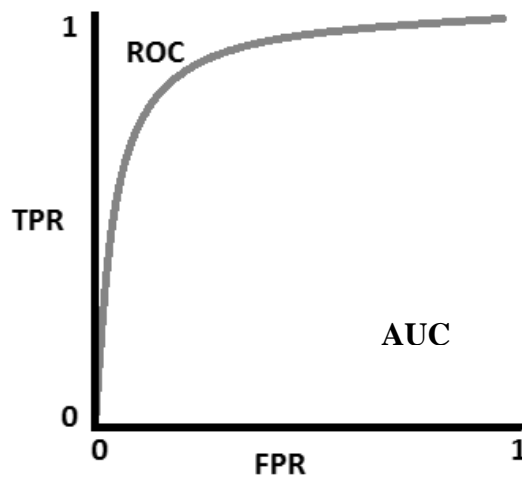
2.8.6. *Loss Function*

Loss function digunakan untuk mengoptimalkan algoritma *machine learning*. *Loss function* dihitung berdasarkan data latih dan data validasi serta interpretasinya didasarkan pada seberapa baik kinerja model dalam dua set ini. *Loss function* adalah jumlah kesalahan yang dibuat untuk setiap contoh dalam *set training* atau validasi. Nilai kerugian(*loss*). menyiratkan seberapa buruk atau baiknya suatu model diterapkan pada setiap iterasi optimasi.

Dalam *deep learning*, *loss* adalah nilai yang coba diminimalkan oleh *Neural Network*. Itulah bagaimana *Neural Network* belajar dengan menyesuaikan bobot dan bias dengan cara mengurangi *loss*. *Training loss* dan *Validation loss* berbeda karena *training loss* diterapkan pada data latih sedangkan *validation loss* diterapkan pada *testing*. Dengan demikian, *validation loss* adalah indikasi yang baik tentang bagaimana kinerja model pada data yang baru dikenali.

2.8.7. Kurva AUC-ROC

Kurva ROC adalah pengukuran kinerja untuk masalah klasifikasi pada berbagai pengaturan ambang batas. *Receiver Operator Characteristic* (ROC) adalah kurva probabilitas dan *Area Under the Curve* (AUC) adalah ukuran yang digunakan sebagai ringkasan dari kurva ROC. Kurva ROC memperlihatkan seberapa besar model mampu membedakan antar kelas. Semakin tinggi AUC, semakin baik model dalam memprediksi 0 sebagai 0 dan 1 sebagai 1. Dengan analogi, semakin tinggi AUC, semakin baik model dalam membedakan antara pasien dengan penyakit dan tidak ada penyakit (Narkhede, 2018).



Gambar 2. 20. Kurva AUC - ROC

Kurva ROC diplotkan dengan *True Positive Rate* (TPR) terhadap *False Positive Rate* (FPR) dimana TPR berada pada sumbu y dan FPR pada sumbu x seperti yang terlihat pada Gambar 2.20.