

**ANALISIS OPTIMIZER PADA CONVOLUTIONAL NEURAL
NETWORK UNTUK MENINGKATKAN AKURASI
PENGENALAN WAJAH**

TUGAS AKHIR

**Hanjaya Suryalim
1118011**



**PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
2022**

**ANALISIS OPTIMIZER PADA CONVOLUTIONAL NEURAL
NETWORK UNTUK MENINGKATKAN AKURASI
PENGENALAN WAJAH**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh
gelar sarjana dalam bidang Informatika**

**Hanjaya Suryalim
1118011**



**PROGRAM STUDI INFORMATIKA
INSTITUT TEKNOLOGI HARAPAN BANGSA
BANDUNG
2022**

ABSTRAK

Nama : Hanjaya Suryalim
Program Studi : Informatika
Judul : Analisis *optimizer* pada *convolutional neural network* untuk meningkatkan akurasi pengenalan wajah

Penggunaan sistem pengenalan wajah semakin meningkat dewasa ini, karena itu penting untuk menemukan cara yang optimal dalam meningkatkan akurasi pengenalan wajah. Pengenalan wajah memanfaatkan teknologi *CNN* yang tersusun dari lapisan-lapisan konvolusi yang diikuti oleh *fully connected layer*. Lapisan konvolusi ini yang bertanggungjawab atas proses ekstraksi fitur pada gambar, yang nantinya digunakan oleh *fully connected layer* untuk mengklasifikasi gambar tersebut. Pada penelitian kali ini diuji dua jenis arsitektur *CNN* yaitu *VGG16* dan *Inception* untuk menguji tingkat akurasi yang dihasilkan. Pengujian juga menggunakan 2 jenis *dataset* yaitu *Komnet* dan *Yale*, serta menguji pengaruh *preprocessing MCLAHE* terhadap hasil akurasi. Selain jenis arsitektur, faktor lain seperti *hyperparameter* juga memegang andil tingkat akurasi model. *Hyperparameter*, yang diuji pada penelitian kali ini adalah jenis *optimizer* dan pengaruh perubahan *learning rate* pada penelitian. *Optimizer* bekerja dengan cara mengubah nilai bobot dan bias saat proses *backpropagation* dengan tujuan menghasilkan nilai *error* yang minimum. Setiap *optimizer* menggunakan algoritma yang unik. Hasil akurasi tertinggi yang dicapai adalah arsitektur *Inception* dengan *optimizer Adadelta* pada *dataset Komnet*. Hasil akurasi pada tahap pelatihan mencapai 98%. Rata-rata akurasi setelah model diuji dengan *10 fold cross validation* adalah 99.3%.

Kata kunci: *CNN, MCLAHE, optimizer, 10 fold cross validation, learning rate, Inception, VGG16.*

ABSTRACT

Name : Hanjaya Suryalim
Department : Informatics
Title : Optimizer Analysis on Convolutional Neural Network for Face Recognition Accuracy Leverage

The usage of face recognition system is arising day by day, that's why it is important to find a way to optimize accuracy rate of the model. Face recognition used CNN as its core technology. CNN consists of multiple layer of convolutional layers followed by fully connected layer. These convolutional layers have a primary function to extract important features from the image where those features will be used in fully connected layer to classify classes. This experiment will take 2 architecture called VGG16 and Inception to test acquired accuracy value using 2 dataset, Komnet and Yale dataset. This experiment also testing the effect of MCLAHE preprocessing on model accuracy. Accuracy of the model also depends on hyperparameter. This experiment will only test on optimizer and learning rate. Optimizer works by updating weight and bias value when backpropagation. Each optimizer has their own unique algorithm. The highest accuracy value in this experiment acquired by the model which trained using komnet dataset and using architecture Inception and adadelata as optimizer. Model acquired accuracy value of 98% in training process. Model also tested using 10 fold cross validation, and its average accuracy is 99.3%.

Keywords: *CNN, MCLAHE, 10 fold cross validation, optimizer, learning rate, Inception, VGG16.*

KATA PENGANTAR

Penulis mengucapkan terima kasih kepada Tuhan Yesus karena dengan karunia-Nya, penulis menyelesaikan Tugas Akhir berjudul "Analisis *optimizer* pada *convolutional neural network* untuk meningkatkan akurasi pengenalan wajah" dengan baik. Penelitian ini disusun sebagai salah satu syarat kelulusan Program Studi Informatika di Institut Teknologi Harapan Bangsa. Lebih dari itu, penulis ingin mencoba mengamalkan bidang keilmuan informatika agar berpartisipasi terhadap kesejahteraan bersama. Perkembangan teknologi sekarang, khususnya *deep learning*, telah memungkinkan pengolahan informasi data gambar yang beragam. Penggunaan teknologi yang tepat guna tentunya sangat bermanfaat untuk membantu pekerjaan manusia yang lebih efisien di masa depan.

Selama penyusunan Tugas Akhir ini, banyak dinamika yang dialami oleh penulis yang tidak dapat dijalani sendirian. Oleh karena itu, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yesus , karena bimbingan dan karunia-Nya, penulis selalu mendapat pengharapan, kebahagiaan, serta pengalaman berharga selama pembuatan Tugas Akhir ini.
2. Ibu Ken Ratri Retno Wardani, S.Kom,M.T., selaku pembimbing utama Tugas Akhir senantiasa memberi dukungan, semangat, waktu bimbingan yang sangat membuka wawasan, ilmu, serta saran yang membangun kepada penulis selama pembuatan Tugas Akhir ini.
3. Dr. Hery Heryanto, M.Kom., selaku pembimbing pendamping Tugas Akhir yang senantiasa memberi dukungan, semangat, ilmu, saran, dan kesempatan menjadi Asisten Dosen selama penyusunan Tugas Akhir ini.
4. Bapak Ventje Jeremias Lewi Engel, S.T., M.T., selaku penguji I Tugas Akhir yang telah memberikan pengujian serta masukan-masukan kritis kepada penulis selama pembuatan Tugas Akhir ini.
5. Bapak Hans Christian Kurniawan, S.T., M.T., selaku penguji II Tugas Akhir yang telah memberikan pengujian serta masukan-masukan kritis kepada penulis selama pembuatan Tugas Akhir ini.
6. Dosen dan *staff* Program Studi Informatika serta DAAK ITHB yang telah membantu menyelesaikan proses administrasi sehingga Tugas Akhir ini diselesaikan.
7. Kedua orang tua yang menyediakan waktu untuk memberikan doa, semangat,

- dan dukungan kepada penulis dalam menyelesaikan Tugas Akhir ini.
8. Annisa Alfa Emmanuel, selaku kekasih yang selalu mendoakan, menemani dan memberi semangat.
 9. Kakak kelas Informatika ITHB, khususnya: Dionisius Pratama, S.Kom. Terima kasih atas inspirasi dan tips menghadapi dan memperjuangkan Tugas Akhir.
 10. Teman teman seperjuangan: Alessandro, Daniel Alexander, Daniel Christianto, Benedictus, Jason Nathanael, dan teman teman lain. Terima kasih untuk waktu bimbingan bersama yang memberikan semangat bagi penulis untuk menyelesaikan Tugas Akhir ini.
 11. Teman teman gereja untuk dukungan doa dan semangat yang diberikan sehingga Tugas Akhir bisa diselesaikan.
 12. Teman seangkatan, adik dan kakak kelas yang berkuliah di satu tempat dan berbeda tempat, yang telah memberikan dukungan, bantuan teknis, semangat, dan waktu diskusi kepada penulis selama melakukan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, baik dari segi konten maupun penulisan, karena keterbatasan waktu, pengetahuan, dan pengalaman yang dimiliki oleh penulis. Penulis menyampaikan permohonan maaf atas hal tersebut. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan penelitian sejenis di masa mendatang.

Akhir kata, semoga Tugas Akhir ini membantu para pembaca yang ingin mengetahui teknologi yang dapat digunakan untuk mengoptimalkan penggunaan *deep learning* dalam bidang sistem pengenalan wajah.

Bandung, 25 Juni 2022

Hormat penulis,



HANJAYA SURYALIM

DAFTAR ISI

ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xvi
DAFTAR ALGORITMA	xxv
DAFTAR LAMPIRAN	xxv
BAB 1 PENDAHULUAN	1-1
1.1 Latar Belakang	1-1
1.2 Rumusan Masalah	1-4
1.3 Tujuan Penelitian	1-4
1.4 Batasan Masalah	1-4
1.5 Kontribusi Penelitian	1-4
1.6 Metodologi Penelitian	1-5
1.7 Sistematika Pembahasan	1-5
BAB 2 LANDASAN TEORI	2-1
2.1 Tinjauan Pustaka	2-1
2.1.1 <i>Neural Network</i>	2-1
2.1.1.1 <i>Perceptron</i>	2-1
2.1.1.2 <i>Deep Learning</i>	2-3
2.1.2 <i>Convolutional Neural Network</i>	2-6
2.1.2.1 <i>Convolution Layer</i>	2-6
2.1.2.2 <i>Pooling Layer</i>	2-13
2.1.2.3 <i>Fully Connected Layer</i>	2-14
2.1.3 <i>Backward Propagation</i>	2-15
2.1.3.1 <i>Gradient Descent</i>	2-17
2.1.3.2 <i>Learning Rate</i>	2-19

2.1.3.3	<i>Cost Function</i>	.2-21
2.1.4	<i>Arsitektur CNN</i>	.2-26
2.1.4.1	<i>VGG-16</i>	.2-27
2.1.4.2	<i>Inception</i>	.2-29
2.1.5	<i>Fungsi Aktivasi</i>	.2-33
2.1.5.1	<i>Rectified Linear Unit (ReLU)</i>	.2-33
2.1.5.2	<i>Softmax</i>	.2-34
2.1.6	<i>Weight Initialization</i>	.2-34
2.1.7	<i>Regularization</i>	.2-36
2.1.7.1	<i>Regularisasi L2</i>	.2-36
2.1.7.2	<i>Batch Normalization Layer</i>	.2-38
2.1.7.3	<i>Dropout Layer</i>	.2-40
2.1.8	<i>Optimizers</i>	.2-41
2.1.8.1	<i>Stochastic Gradient Descent</i>	.2-42
2.1.8.2	<i>Nesterov Accelerated Gradient (NAG)</i>	.2-45
2.1.8.3	<i>Adagrad</i>	.2-47
2.1.8.4	<i>Adadelata</i>	.2-48
2.1.8.5	<i>Adam</i>	.2-50
2.1.9	<i>Image Preprocessing</i>	.2-52
2.1.9.1	<i>Image Interpolation</i>	.2-52
2.1.9.2	<i>Aspect Aware Preprocessing</i>	.2-53
2.1.9.3	<i>Image Augmentation</i>	.2-54
2.1.9.4	<i>LAB - colorspace</i>	.2-62
2.1.9.5	<i>Histogram Equalization (HE)</i>	.2-64
2.1.9.6	<i>Adaptive Histogram Equalization (AHE)</i>	.2-65
2.1.9.7	<i>Contrast Limited Adaptive Histogram Equalization (CLAHE)</i>	.2-65
2.1.9.8	<i>Modified Contrast Limited Adaptive Histogram Equalization (MCLAHE)</i>	.2-67
2.1.9.9	<i>Gaussian Blur</i>	.2-68
2.1.10	<i>Confusion Matrix</i>	.2-69
2.1.10.1	<i>Macro Average</i>	.2-71
2.1.10.2	<i>Weighted Average</i>	.2-72
2.1.10.3	<i>Micro Average</i>	.2-72
2.1.11	<i>Cross Validation</i>	.2-72
2.1.12	<i>Grad-CAM</i>	.2-73
2.1.13	<i>Pustaka</i>	.2-74

2.1.13.1	TensorFlow	2-74
2.1.13.2	Keras	2-76
2.1.13.3	OpenCV	2-83
2.1.13.4	NumPy	2-85
2.1.13.5	OS	2-85
2.1.13.6	Matplotlib	2-86
2.1.13.7	Imutils	2-87
2.1.13.8	Sklearn	2-88
2.2	Tinjauan Studi	2-89
2.2.1	<i>State of the Art</i>	2-90
2.2.2	Pembahasan penelitian terkait	2-91
2.3	Tinjauan Objek	2-91
2.3.1	<i>Cropped Extended Yale Face Database</i>	2-92
2.3.2	<i>Komnet Dataset</i>	2-93
BAB 3	ANALISIS DAN PERANCANGAN SISTEM	3-1
3.1	Analisis Masalah	3-1
3.2	Kerangka Pemikiran	3-2
3.3	Analisis Urutan Proses Global	3-5
3.3.1	Pemrosesan awal	3-6
3.3.2	Proses Pelatihan	3-8
3.3.2.1	<i>Forward Propagation Convolutional Neural Network</i>	3-9
3.3.2.2	<i>Backward Propagation Convolutional Neural Network</i>	3-11
3.3.3	Proses Pengujian	3-12
3.4	Analisis Manual	3-13
3.4.1	Analisis <i>Dataset</i>	3-13
3.4.1.1	<i>Aspect Aware Preprocessor & Image to Array preprocessor</i>	3-13
3.4.1.2	<i>Modified Contrast Limited Adaptive Histogram Equalization</i>	3-14
3.4.1.3	Augmentasi Dataset	3-14
3.4.2	<i>Forward Propagation</i>	3-15
3.4.2.1	<i>Forward Propagation VGG16</i>	3-15
3.4.2.2	<i>Forward Propagation Inception</i>	3-24
3.4.3	<i>Backward Propagation</i>	3-32

3.4.3.1	<i>Backward propagation pada Stochastic Gradient Descent</i>	3-32
3.4.3.2	<i>Backward propagation pada Nesterov Accelerated Gradient</i>	3-32
3.4.3.3	<i>Backward propagation pada Adagrad</i>	3-33
3.4.3.4	<i>Backward propagation pada Adadelta</i>	3-34
3.4.3.5	<i>Backward propagation pada Adam</i>	3-35
BAB 4	IMPLEMENTASI DAN PENGUJIAN	4-1
4.1	Lingkungan Implementasi	4-1
4.1.1	Spesifikasi Perangkat Keras	4-1
4.1.2	Lingkungan Perangkat Lunak	4-1
4.2	Implementasi Perangkat Lunak	4-1
4.2.1	Penggunaan <i>Jupyter Lab</i>	4-1
4.2.2	Penggunaan <i>Google Colaboratory</i>	4-2
4.2.2.1	Class VGG16	4-3
4.2.2.2	Class Inception	4-3
4.2.2.3	Class ImageToArrayPreprocessor	4-5
4.2.2.4	Class AspectAwarePreprocessor	4-5
4.2.2.5	Class DatasetLoader	4-5
4.2.3	Implementasi Penggunaan <i>Dataset</i>	4-6
4.3	Implementasi Aplikasi	4-7
4.4	Pengujian	4-9
4.4.1	Pengujian <i>Benchmark Dataset</i>	4-9
4.4.2	Pengujian <i>Preprocessing</i>	4-9
4.4.3	Pengujian Jenis Arsitektur <i>Neural Network</i>	4-9
4.4.4	Pengujian Jenis <i>Optimizer</i>	4-9
4.4.5	Pengujian <i>Learning Rate tuning</i>	4-10
4.5	Hasil Pengujian	4-10
4.5.1	Pengujian pada dataset <i>Original Cropped Yale</i>	4-10
4.5.1.1	Pengujian <i>Cropped Yale</i> pada arsitektur VGG16	4-11
4.5.1.2	Pengujian <i>Cropped Yale</i> pada arsitektur <i>Inception</i>	4-17
4.5.1.3	Tabel Hasil Pengujian pada dataset <i>Original Yale</i>	4-22
4.5.2	Pengujian pada dataset <i>MCLAHE Cropped Yale</i>	4-27
4.5.2.1	Pengujian <i>MCLAHE Cropped Yale</i> pada arsitektur VGG16	4-27
4.5.2.2	Pengujian <i>MCLAHE Cropped Yale</i> pada arsitektur <i>Inception</i>	4-33

4.5.2.3	Tabel Hasil Pengujian pada dataset <i>MCLAHE Yale</i>	4-38
4.5.3	Pengujian pada dataset <i>Original Komnet</i>	4-43
4.5.3.1	Pengujian Komnet pada arsitektur VGG16	4-43
4.5.3.2	Pengujian Komnet pada arsitektur <i>Inception</i>	4-49
4.5.3.3	Tabel Hasil Pengujian pada dataset Komnet	4-54
4.5.4	Pengujian pada dataset <i>MCLAHE Komnet</i>	4-59
4.5.4.1	Pengujian <i>MCLAHE</i> Komnet pada arsitektur VGG16	4-59
4.5.4.2	Pengujian <i>MCLAHE</i> Komnet pada arsitektur <i>Inception</i>	4-65
4.5.4.3	Tabel Hasil Pengujian pada dataset <i>Komnet MCLAHE</i>	4-70
4.5.5	Hasil Pengujian akurasi dengan <i>learning rate tuning</i>	4-76
4.5.5.1	Pengujian pada Dataset <i>Yale</i>	4-77
4.5.5.2	Pengujian pada Dataset <i>Yale MCLAHE</i>	4-84
4.5.5.3	Pengujian pada Dataset <i>Komnet</i>	4-92
4.5.5.4	Pengujian pada Dataset <i>Komnet MCLAHE</i>	4-100
4.5.6	Pembahasan Hasil Pengujian <i>Optimizer</i> dan <i>Learning rate tuning</i>	4-109
4.6	Analisis Kesalahan	4-124
BAB 5	KESIMPULAN DAN SARAN	5-1
5.1	Kesimpulan	5-1
5.2	Saran	5-2
DAFTAR REFERENSI		i
LAMPIRAN A		
DATASET YALE & KOMNET		A-1
LAMPIRAN B		
Hasil Pengujian Arsitektur <i>Convolutional Neural Network</i>		B-2

DAFTAR TABEL

2.1	<i>Confusion matrix</i>	2-70
2.2	Daftar metode yang digunakan dari pustaka TensorFlow	2-75
2.3	Daftar metode yang digunakan dari pustaka Keras	2-77
2.4	Daftar metode yang digunakan dari pustaka OpenCV	2-83
2.5	Daftar metode yang digunakan dari pustaka NumPy	2-85
2.6	Daftar metode yang digunakan dari pustaka OS	2-86
2.7	Daftar metode yang digunakan dari pustaka matplotlib	2-87
2.8	Daftar <i>method</i> yang digunakan dari pustaka imutils	2-88
2.9	Daftar metode yang digunakan dari pustaka sklearn	2-88
2.10	<i>State of the Art Method</i>	2-90
2.10	<i>State of the Art Method</i>	2-91
3.1	Contoh matriks citra	3-17
3.2	Contoh matriks citra dengan <i>padding</i>	3-17
3.3	Contoh nilai <i>kernel</i> 3×3	3-18
3.4	Contoh nilai <i>feature map</i> hasil konvolusi	3-19
3.5	Contoh nilai <i>feature map</i> setelah diaktivasi	3-19
3.6	Contoh <i>feature map</i> hasil perhitungan <i>batch normalization</i>	3-20
3.7	Contoh hasil perhitungan <i>max pooling</i>	3-21
3.8	Contoh <i>feature tensor</i> 1 dimensi	3-21
3.9	Contoh <i>probability dropout</i>	3-22
3.10	<i>dropout result</i>	3-22
3.11	Contoh <i>kernel</i> pada lapisan <i>dense</i>	3-22
4.1	Daftar metode pada <i>Jupyter lab</i>	4-2
4.2	Daftar metode pada <i>class VGG16</i>	4-3
4.3	Daftar <i>method</i> pada <i>class Inception</i>	4-4
4.4	Daftar <i>method</i> pada <i>class ImageToArrayPreprocessor</i>	4-5
4.5	Daftar <i>method</i> pada <i>class AspectAwarePreprocessor</i>	4-5
4.6	Daftar <i>method</i> pada <i>class DatasetLoader</i>	4-6
4.7	Perincian Penggunaan <i>Dataset</i> untuk Implementasi	4-6
4.8	Hasil pengujian (<i>accuracy</i>) pada arsitektur <i>VGG16 & Inception</i>	4-23
4.9	Tabel perbandingan (<i>accuracy</i>) pada arsitektur <i>VGG16</i> dan <i>Inception</i>	4-25
4.10	Hasil pengujian (<i>accuracy</i>) pada arsitektur <i>VGG16 & Inception</i>	4-39

4.11	Tabel perbandingan (<i>accuracy</i>) pada arsitektur <i>VGG16</i> dan <i>Inception</i>	4-41
4.12	Hasil pengujian (<i>accuracy</i>) pada arsitektur <i>VGG16</i> dan <i>Inception</i> .	4-55
4.13	Tabel perbandingan (<i>accuracy</i>) pada arsitektur <i>VGG16</i> dan <i>Inception</i>	4-57
4.14	Hasil pengujian (<i>accuracy</i>) pada arsitektur <i>VGG16</i> dan <i>Inception</i> .	4-71
4.15	Tabel perbandingan (<i>accuracy</i>) pada arsitektur <i>VGG16</i> dan <i>Inception</i>	4-73
4.16	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale</i> dengan arsitektur <i>VGG16</i>	4-77
4.17	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale</i> dengan arsitektur <i>Inception</i>	4-81
4.18	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale MCLAHE</i> dengan arsitektur <i>VGG16</i>	4-85
4.19	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale MCLAHE</i> dengan arsitektur <i>Inception</i>	4-89
4.20	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet</i> dengan arsitektur <i>VGG16</i>	4-93
4.21	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet</i> dengan arsitektur <i>Inception</i>	4-97
4.22	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet MCLAHE</i> dengan arsitektur <i>VGG16</i>	4-101
4.23	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet MCLAHE</i> dengan arsitektur <i>Inception</i>	4-105
4.24	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale</i> dengan arsitektur <i>VGG16</i>	4-111
4.25	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale</i> dengan arsitektur <i>Inception</i>	4-112
4.26	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale MCLAHE</i> dengan arsitektur <i>VGG16</i>	4-112
4.27	Pengujian <i>learning rate tuning</i> pada dataset <i>Yale MCLAHE</i> dengan arsitektur <i>Inception</i>	4-113
4.28	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet</i> dengan arsitektur <i>VGG16</i>	4-114
4.29	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet</i> dengan arsitektur <i>Inception</i>	4-114

4.30	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet MCLAHE</i> dengan arsitektur <i>VGG16</i>	4-115
4.31	Pengujian <i>learning rate tuning</i> pada dataset <i>Komnet MCLAHE</i> dengan arsitektur <i>Inception</i>	4-116
A	Gambar dan nama kelas <i>Yale dataset</i>	A-1
A	Gambar dan nama kelas <i>Yale dataset</i>	A-2
A	Gambar dan nama kelas <i>Yale dataset</i>	A-3
A	Gambar dan nama kelas <i>Yale dataset</i>	A-4
A	Gambar dan nama kelas <i>Yale dataset</i>	A-5
A	Gambar dan nama kelas <i>Yale dataset</i>	A-6
A	Gambar dan nama kelas <i>Yale dataset</i>	A-7
A	Gambar dan nama kelas <i>Yale dataset</i>	A-8
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-9
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-10
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-11
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-12
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-13
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-14
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-15
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-16
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-17
B	Gambar dan nama kelas <i>Komnet dataset</i>	A-1
A	Hasil akurasi pengujian <i>Yale dataset</i>	B-2
A	Hasil akurasi pengujian <i>Yale dataset</i>	B-3
B	Hasil akurasi pengujian <i>Yale MCLAHE dataset</i>	B-3
B	Hasil akurasi pengujian <i>Yale MCLAHE dataset</i>	B-4
C	Hasil akurasi pengujian dataset <i>Komnet</i>	B-4
C	Hasil akurasi pengujian dataset <i>Komnet</i>	B-5
D	Hasil akurasi pengujian dataset <i>Komnet MCLAHE</i>	B-5
D	Hasil akurasi pengujian dataset <i>Komnet MCLAHE</i>	B-6

DAFTAR GAMBAR

2.1	<i>Perceptron</i> [9]	2-2
2.2	<i>Multi Layer Perceptron</i> [8]	2-3
2.3	<i>Deep Neural Network</i> [14]	2-4
2.4	Contoh citra uji memuat gambar kangguru, anak manusia dan bintang [14]	2-4
2.5	<i>Deep Learning Schema</i> [14]	2-5
2.6	<i>Deep Learning Schema</i> [14]	2-5
2.7	<i>Convolution Layer</i> [14]	2-7
2.8	<i>Convolution Layer with strides</i> [15]	2-7
2.9	<i>Strides</i> [14]	2-8
2.10	<i>Depth in convolutional layer</i> [15]	2-9
2.11	<i>convolution process</i> [37]	2-10
2.12	<i>convolution process lanjutan</i> [37]	2-11
2.13	<i>Enlargement convolution process</i> [37]	2-11
2.14	<i>Hierarki CNN</i> [14]	2-13
2.15	<i>Pooling Layer</i> [6]	2-14
2.16	<i>Fully Connected Layer</i> [15]	2-15
2.17	<i>Backward Propagation</i>	2-16
2.18	Contoh <i>Gradient Descent</i> [10]	2-18
2.19	<i>Constant learning rate</i> [22]	2-19
2.20	<i>Using learning rate decay</i> [22]	2-20
2.21	<i>Cross Entropy</i> [12]	2-26
2.22	<i>Convolutional Neural Network</i> [15]	2-26
2.23	Struktur VGG16 [20]	2-28
2.24	Lapisan VGG16 [20]	2-28
2.25	Perbandingan arsitektur VGG [19]	2-29
2.26	<i>Inception problem</i> [20]	2-29
2.27	<i>Convolution module</i>	2-30
2.28	<i>Inception module</i>	2-31
2.29	<i>Inception module architecture</i> [20]	2-31
2.30	<i>Normal Convolution</i> [20]	2-32
2.31	<i>1x1 convolution layer</i> [20]	2-32
2.32	ReLU [30]	2-33
2.33	<i>derivative of ReLU</i> [30]	2-33

2.34	<i>Neural Network</i>	2-35
2.35	<i>Dropout layer</i> [36]	2-40
2.36	Pergerakan grafik <i>cost fuction</i> mencapai titik minimum tahap 1 [10]	2-42
2.37	Pergerakan grafik <i>cost fuction</i> mencapai titik minimum tahap 2 [10]	2-43
2.38	Pergerakan grafik <i>cost fuction</i> mencapai titik minimum tahap 3 [10]	2-43
2.39	<i>Stochastic Gradient Descent</i> [10]	2-44
2.40	<i>Mini Batch Gradient Descent</i> [10]	2-45
2.41	<i>Local Minimum</i> [26]	2-45
2.42	<i>NAG movement</i> [26]	2-47
2.43	<i>Aspect Aware Preprocess</i> [25]	2-53
2.44	<i>rotation process</i> [42]	2-56
2.45	Contoh gambar yang dirotasi [39]	2-56
2.46	Proses <i>shift</i> [39]	2-57
2.47	Proses <i>flip</i> [39]	2-57
2.48	Proses <i>brighthness</i> [39]	2-58
2.49	<i>zoom process</i> [39]	2-58
2.50	<i>Original Image</i>	2-58
2.51	<i>Row wised zoom</i>	2-58
2.52	<i>Column wised zoom</i>	2-59
2.53	<i>Original Image</i>	2-59
2.54	<i>Row wised zoom</i>	2-59
2.55	<i>Column wised zoom</i>	2-59
2.56	<i>gambar original</i>	2-60
2.57	<i>K-times zoom 1</i>	2-60
2.58	<i>K-times zoom 2</i>	2-60
2.59	<i>Column wised zoom</i>	2-60
2.60	<i>Shear x example</i>	2-61
2.61	<i>LAB channel example</i>	2-64
2.62	<i>Before Histogram Equalization</i> [40]	2-64
2.63	<i>After Histogram Equalization</i> [40]	2-64
2.64	<i>Contrast Limited Adaptive Histogram Equalization (CLAHE)</i> [1]	2-66
2.65	<i>threshold cut in CLAHE</i> [1]	2-67
2.66	<i>Gaussian Kernel 5x5</i>	2-69
2.67	<i>Macro Average</i> [21]	2-71
2.68	<i>Weighted Average</i> [21]	2-72
2.69	<i>Micro Average</i> [21]	2-72
2.70	<i>Grad CAM Schema</i> [45]	2-73

2.71	<i>Cropped Yale Face Database Illumination</i>	2-92
2.72	<i>Komnet Dataset example</i>	2-93
3.1	Kerangka Pemikiran	3-2
3.2	<i>Flowchart</i> urutan proses global	3-5
3.3	<i>Preprocessing dataset</i>	3-6
3.4	<i>Flowchart</i> pelatihan sistem pengenalan wajah	3-8
3.5	arsitektur VGG16	3-16
3.6	arsitektur <i>Inception</i>	3-25
4.1	<i>Homepage</i>	4-7
4.2	<i>Homepage - Upload</i>	4-8
4.3	<i>Homepage -Predict</i>	4-8
4.4	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 tanpa menggunakan <i>optimizer</i>	4-11
4.5	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 tanpa menggunakan <i>optimizer</i>	4-11
4.6	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-12
4.7	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-12
4.8	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-13
4.9	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-13
4.10	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-14
4.11	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-14
4.12	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-15
4.13	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-15
4.14	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-16
4.15	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-16

4.16	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-17
4.17	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-17
4.18	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-18
4.19	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-18
4.20	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-19
4.21	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-19
4.22	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-20
4.23	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-20
4.24	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-21
4.25	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-21
4.26	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-22
4.27	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-22
4.28	<i>VGG16 accuracy & validation accuracy</i>	4-24
4.29	<i>Inception accuracy & validation accuracy</i>	4-25
4.30	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> tanpa menggunakan <i>optimizer</i>	4-27
4.31	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> tanpa menggunakan <i>optimizer</i>	4-27
4.32	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-28
4.33	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-28

4.34	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-29
4.35	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-29
4.36	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-30
4.37	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-30
4.38	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adadelata</i>	4-31
4.39	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adadelata</i>	4-31
4.40	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-32
4.41	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-32
4.42	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-33
4.43	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-33
4.44	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-34
4.45	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-34
4.46	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-35
4.47	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-35
4.48	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-36
4.49	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-36

4.50	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-37
4.51	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-37
4.52	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-38
4.53	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-38
4.54	<i>VGG16 accuracy & validation accuracy</i>	4-40
4.55	<i>Inception accuracy & validation accuracy</i>	4-41
4.56	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> tanpa menggunakan <i>optimizer</i>	4-43
4.57	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> tanpa menggunakan <i>optimizer</i>	4-43
4.58	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-44
4.59	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-44
4.60	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-45
4.61	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-45
4.62	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-46
4.63	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-46
4.64	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-47
4.65	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-47
4.66	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-48
4.67	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-48

4.68	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-49
4.69	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-49
4.70	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-50
4.71	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-50
4.72	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-51
4.73	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-51
4.74	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-52
4.75	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-52
4.76	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelata</i>	4-53
4.77	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelata</i>	4-53
4.78	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-54
4.79	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-54
4.80	<i>VGG16 accuracy & validation accuracy</i>	4-56
4.81	<i>Inception accuracy & validation accuracy</i>	4-57
4.82	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> tanpa menggunakan <i>optimizer</i>	4-59
4.83	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> tanpa menggunakan <i>optimizer</i>	4-59
4.84	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-60
4.85	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>VGG16</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-60

4.86	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-61
4.87	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-61
4.88	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-62
4.89	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-62
4.90	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adadelata</i>	4-63
4.91	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adadelata</i>	4-63
4.92	Grafik perubahan akurasi validasi pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-64
4.93	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur VGG16 dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-64
4.94	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-65
4.95	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> tanpa menggunakan <i>optimizer</i>	4-65
4.96	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-66
4.97	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Stochastic Gradient Descent</i>	4-66
4.98	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-67
4.99	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Nesterov Accelerated Gradient</i>	4-67
4.100	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-68
4.101	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adagrad</i>	4-68

4.102	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-69
4.103	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adadelta</i>	4-69
4.104	Grafik perubahan akurasi validasi pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-70
4.105	Grafik perubahan nilai <i>loss</i> pengujian pada arsitektur <i>Inception</i> dengan menggunakan <i>optimizer</i> berupa <i>Adam</i>	4-70
4.106	<i>VGG16 accuracy & validation accuracy</i>	4-72
4.107	<i>Inception accuracy & validation accuracy</i>	4-73
4.108	<i>Yale - VGG16</i>	4-75
4.109	<i>Yale - VGG16</i>	4-75
4.110	<i>Yale - Inception</i>	4-75
4.111	<i>Komnet - Inception</i>	4-76
4.112	<i>VGG16</i>	4-116
4.113	<i>Inception</i>	4-117
4.114	<i>VGG16 model accuracies</i>	4-118
4.115	<i>Inception model accuracies</i>	4-118
4.116	<i>gradcam VGG16</i>	4-120
4.117	<i>gradcam VGG16</i>	4-121
4.118	<i>gradcam inception</i>	4-122
4.119	<i>gradcam inception</i>	4-123
4.120	Uji coba aplikasi 1	4-125
4.121	Uji coba aplikasi 2	4-125

DAFTAR ALGORITMA

Algoritma 2.1	<i>algoritma Aspect Aware Preprocessing</i>	2-53
Algoritma 2.2	<i>K-times zooming algorithm [42]</i>	2-60
Algoritma 2.3	<i>Conversion between RGB to LAB channel</i>	2-63
Algoritma 2.4	<i>HE Algorithm [40]</i>	2-64
Algoritma 2.5	<i>MCLAHE algorithm</i>	2-68
Algoritma 3.1	<i>Forward Propagation Convolutional Neural Network</i> . .	3-10
Algoritma 3.2	<i>Backward Propagation Convolutional Neural Network</i> <i>without optimizer</i>	3-11
Algoritma 3.3	<i>Backward Propagation Convolutional Neural Network</i> <i>with optimizer</i>	3-12

DAFTAR LAMPIRAN

LAMPIRAN A	A-1
LAMPIRAN B	B-2

DAFTAR REFERENSI

- [1] Bendjilali Ridha Ilyas, Beladgham Mohammed, Merit Khaled and Taleb-Ahmed Abdelmalik, *"Illumination-robust face recognition based on deep convolutional neural networks architecture"*, *Indonesian Journal of Electrical Engineering and Computer Science*, May 2020.
- [2] Brownlee, Jason PhD. *Why Optimization Is Important in Machine Learning*, <https://machinelearningmastery.com/why-optimization-is-important-in-machine-learning/> (accessed February 12, 2022)
- [3] Yaqub Muhammad, Feng Jinchao, Zia Sultan, Arshid Kaleem, Jia Kebin, Rehman Zaka Ur adn Mehmood Atif *"State-of-the-Art CNN Optimizer for Brain Tumor Segmentation in Magnetic Resonance Images"*, *Brain Sciences*, published July 3, 2020.
- [4] Samar S. Mohammed, Wael A. Mohamed, A.T. Khalil and A.S. Mora, *"Deep Learning Face Detection and Recognition"*, *Iternational journal of electronics and telecommunications*, June 2019.
- [5] KB. Pranav, J. Manikandan, *"Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks"*, *Article in Procedia Computer Science 171 (2020) 1651–1659*, 2020.
- [6] Coskun Musab, Ucar Aysegul, Yildirm Ozal and Demir Yakup, *"Face Recognition Based on Convolutional Neural Network"*, *Conference Paper*, November, 2021.
- [7] Zhiming Xie, Junjie Li and Hui Shi, *"A Face Recognition Method Based on CNN"*, *2019 High Performance Computing and Computational Intelligence Conference*, 2019
- [8] IBM Cloud Education. *Neural Networks*. <https://www.ibm.com/cloud/learn/neural-networks> (accessed September 13, 2021 17:17:23)
- [9] Bhardwaj, Anjali. *What is a Perceptron? – Basics of Neural Networks*. <https://towardsdatascience.com/what-is-a-perceptron-basics-of-neural-networks-c4cfea20c590> (accessed September 13, 2021 12:30:34)

DAFTAR REFERENSI

- [10] Donges, Niklas. *Gradient Descent: An Introduction to 1 of Machine Learning's Most Popular Algorithms*. <https://builtin.com/data-science/gradient-descent> (accessed September 14, 2021 11:40:23)
- [11] Pandey, Paul. *Understanding the Mathematics behind Gradient Descent*. <https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e> (accessed September 14, 2021 11:46:30)
- [12] Shah, Saily. *Cost Function is No Rocket Science!* <https://www.analyticsvidhya.com/blog/2021/02/cost-function-is-no-rocket-science/> (accessed September 14, 2021 12:45:12)
- [13] B.Dickson. *"What are convolutional neural networks (CNN)?"*.bdtechtalk.com. <https://bdtechtalks.com/2020/01/06/convolutional-neural-networks-cnn-convnets/> (accessed May 4, 2021 14:55:23)
- [14] IBM Cloud Education. *Convolutional Neural Networks* <https://www.ibm.com/cloud/learn/convolutional-neural-networks> (accessed October 25, 2021 13:30:30)
- [15] Jeong, Jiwon. *The Most Intuitive and Easiest Guide for Convolutional Neural Network* <https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480> (accessed October 25, 2021 13:36 54)
- [16] Saxena, Shipra. *Introduction to Batch Normalization* <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-batch-normalization/> (accessed October 25, 2021 17:43:42)
- [17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning, An MIT Press book*, 2016. [Online]. Available : <http://www.deeplearningbook.org> (accessed November 7, 2021 16:57:00)
- [18] Mustafa. *Optimizers in Deep Learning* <https://medium.com/mllearning-ai/optimizers-in-deep-learning-7bf81fed78a0> (accessed October 31, 2021 19:49:50)
- [19] Simonyan. Karen and Zisserman. Andrew, *"VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE SCALE IMAGE RECOGNITION" conference paper at ICLR 2015, April 10 2015*

DAFTAR REFERENSI

- [20] Sharma, Pulkit. "A Comprehensive Tutorial to learn Convolutional Neural Networks from Scratch (deeplearning.ai Course #4)". <https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/> (accessed October 25, 2021 23:54:50)
- [21] Leung, Kenneth. "Micro, Macro Weighted Averages of F1 Score, Clearly Explained". <https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f> (accessed March 3, 2022 13:31:00)
- [22] Haswani, Vaibhav. "Learning Rate Decay and methods in Deep Learning". <https://medium.com/analytics-vidhya/learning-rate-decay-and-methods-in-deep-learning-2cee564f910b>: :text=Learning rate decay is a,help both optimization and generalization (accessed March 3, 2022 13:17:00)
- [23] Kumar, Ahlad. "Lecture: CNN Architectures (AlexNet, VGGNet, Inception ResNet)". <https://www.youtube.com/watch?v=CNnNzl8HIIU> (accessed October 26, 2021 13:24:20)
- [24] Rosebrock, Adrian. 2017. *Deep Learning for Computer Vision with Python*. PyImageSearch
- [25] Rosebrock, Adrian. 2017. *Deep Learning for Computer Vision with Python*. pp. 20 - 22.
- [26] Ruder, Sebastian. 2016. *An overview of gradient descent optimization algorithms*. <https://www.notion.so/Gradient-with-momentum-b1b93b72ff71402ebd6c47b49a408f81> (accessed November 1, 2021 (16:28:00))
- [27] Kumawat, Neha. *Continuing on Adaptive Method: ADADELTA and RMSProp*. <https://insideaiml.com/blog/Adagrad-and-Adadelta-optimizer:-In-depth-explanation-1052> (accessed January 9, 2022 (21:42:00))
- [28] Huilgol, Purva. *Accuracy vs. F1-Score*. <https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2>: :text=Accuracy is used when the, and False Positives are crucial text=In most real life classification, to evaluate our model on (accessed February 12 2022, 22:39:00)
- [29] Ilyas, Ridwan. *Penjelasan Algoritma Backpropagation Mengupdate Bobot | Jurnal Kelas*. <https://www.youtube.com/watch?v=OeowRowEbBc> (accessed November 7, 2021 11:41:00)

DAFTAR REFERENSI

- [30] Gupta, Dishashree. *Fundamentals of Deep Learning – Activation Functions and When to Use Them?*.<https://www.analyticsvidhya.com/blog/2020/01/fundamentals-deep-learning-activation-functions-when-to-use-them/> (accessed November 7, 2021 11:45:00)
- [31] Naik, Krish. *Deep Learning-All Optimizers In One Video-SGD with Momentum,Adagrad,Adadelta,RMSprop,Adam Optimizers*.<https://www.youtube.com/watch?v=TudQZtgpoHk> (accessed November 7, 2021 13:08:00)
- [32] opencv. *Color conversions*. https://docs.opencv.org/3.4.15/de/d25/imgproc_color_conversions.html (accessed November 7, 2021 15:26:00)
- [33] S. Khan, H. Rahmani, Syed Afaq Ali Shah, M. Bennamoun, "A Guide to Convolutional Neural Networks for Computer Vision", 1sted, GrardMedioniandSvenDickinson, Ed.California : MorganandClaypool, 2018, pp.45, 53, 56, 67 – 80.
- [34] Kapoor, Namrata. *Weight Initialization Techniques-What best works for you*.<https://www.numpyninja.com/post/weight-initialization-techniques> (accessed November 7, 2021 20:15:00)
- [35] Taunk, Dhaval. *L1 vs L2 Regularization: The intuitive difference*.<https://medium.com/analytics-vidhya/l1-vs-l2-regularization-which-is-better-d01068e6658c> (accessed November 7, 2021 21:40:00)
- [36] H. Habibi Aghdam and E. Jahani Heravi, *Guide to Convolutional Neural Networks*, 1 st ed. Switzerland: Springer International Publishing AG, 2017, pp. 108-111, 118-120.
- [37] Lina, Qobylatul. *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Gambar (Mata Juling dan Mata Normal) dengan R*. <https://medium.com/@16611110/implementasi-deep-learning-menggunakan-convolutional-neural-network-untuk-klasifikasi-gambar-mata-87dcc0ad26e0> (accessed November 16, 2021 22:39:00)
- [38] H. Kinsley and D. Kukiela, "Neural Networks from Scratch in Python", 1 st ed. Harrison Kinsley, 2020, pp. 108, 333-358.
- [39] Bhandari Aniruddha, *Image Augmentation on the fly using Keras ImageDataGenerator*.

DAFTAR REFERENSI

- <https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/> (accessed November 24, 2021 18:23:00)
- [40] Sudhakar, Shreenidhi. *Histogram Equalization*. <https://towardsdatascience.com/histogram-equalization-5d1013626e64> (accessed November 25, 2021 12:16:00)
- [41] Frickie, Tobin. *Rotation by Shearing*. https://www.ocf.berkeley.edu/~fricke/projects/israel/paeth/rotation_by_shearing.html (accessed November 27, 2021 12:15:00)
- [42] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd ed. New Jersey: Pearson Prentice Hall, 2008, pp.91, pp.122-126
- [43] H. Habibi Aghdam and E. Jahani Heravi, *Guide to Convolutional Neural Networks*, 1st ed. Switzerland: Springer International Publishing AG, 2017, pp. 108-111, 118-120.
- [44] Naik Krish, *Tutorial 14 - Stochastic Gradient Descent with Momentum*, <https://www.youtube.com/watch?v=CKLwvuKWQjo> (accessed February 7. 2022 23:41:00)
- [45] Prof Balasubramanian Vineeth, *Explaining CNNs: Class Attribution Map Methods*, <https://www.youtube.com/watch?v=VmbBnSv3otc> (accessed February 7. 2022 22:00:00)