



TUGAS AKHIR - EE 184801

**ANALISIS PERFORMA SISTEM DETEKSI OBJEK KONSTRUKSI
BANGUNAN MENGGUNAKAN *CONVOLUTIONAL NEURAL
NETWORK***

Ahmad Athfi Noor Laili
NRP 07111540007004

Dosen Pembimbing
Prof. Dr. Ir. Achmad Jazidie, M.Eng.
Imam Arifin, ST., MT.
Mahardhika Pratama, M.Sc., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



FINAL PROJECT - EE 184801

**PERFORMANCE ANALYSIS OF BUILDING CONSTRUCTION OBJECT
DETECTION SYSTEM USING CONVOLUTIONAL NEURAL
NETWORK**

Ahmad Athfi Noor Laili
NRP 07111540007004

Supervisor
Prof. Dr. Ir. Achmad Jazidie, M.Eng.
Imam Arifin, ST., MT.
Mahardhika Pratama, M.Sc., Ph.D.

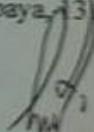
DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Analisis Performa Sistem Deteksi Objek Konstruksi Bangunan Menggunakan *Convolutional Neural Network***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 13 Januari 2020



Ahmad Athfi Noor Laili
NRP. 07111540007004

Halaman ini sengaja dikosongkan

**ANALISIS PERFORMA SISTEM DETEKSI OBJEK
KONSTRUKSI BANGUNAN MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro**

Pada

**Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Prof. Dr. Ir. Achmad Jazidie, M.Eng.

Imam Arifin, ST., MT.

NIP. 195902191986101001

NIP.197302222002121001

Dosen Pembimbing III

Mahardhika Pratama, M.Sc., Ph.D.

**Asst. Prof School of Computer Science and Engineering
Nanyang Technological University, Singapore**

**SURABAYA
DESEMBER, 2020
DEPARTEMEN
TEKNIK ELEKTRO**

Halaman ini sengaja dikosongkan

ANALISIS PERFORMA SISTEM DETEKSI OBJEK KONSTRUKSI BANGUNAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK

Ahmad Athfi Noor Laili
07111540007004

Dosen Pembimbing : 1. Prof. Dr. Ir. Achmad Jazidie, M.Eng.
: 2. Imam Arifin, ST., MT.
: 3. Mahardhika Pratama, M.Sc., Ph.D.

ABSTRAK

Deep learning merupakan salah satu cabang dari *machine learning* yang memungkinkan komputer dapat belajar dari data dan pengalaman. Salah satu algoritma dalam *deep learning* adalah *Convolutional Neural Network* (CNN) dimana sering digunakan dalam deteksi objek. Namun, terdapat beberapa kendala antara lain jika objek pada gambar yang dideteksi terhalang objek lainnya, akurasi yang dihasilkan mengalami penurunan dan resolusi yang rendah dapat mengakibatkan objek tidak terdeteksi. Oleh karena itu, dalam penelitian ini digunakan metode Faster R-CNN untuk meningkatkan performa sistem deteksi objek dengan menggunakan dua model CNN yakni Resnet50 dan VGG-16. Hasil dari pengujian yang diperoleh dengan menggunakan model ResNet50 mempunyai kesalahan 0,3471 dengan tingkat akurasi sebesar 85,33%. Sedangkan hasil pengujian dengan model VGG-16 memiliki kesalahan sebesar 0,6782 dengan akurasi sebesar 93,39%.

Kata Kunci : *Convolutional neural network*, klasifikasi gambar, ResNet50, VGG-16.

Halaman ini sengaja dikosongkan

**PERFORMANCE ANALYSIS OF BUILDING CONSTRUCTION
OBJECT DETECTION SYSTEM USING CONVOLUTIONAL NEURAL
NETWORK**

Ahmad Athfi Noor Laili
07111540007004

Supervisor : 1. Prof. Dr. Ir. Achmad Jazidie, M.Eng.
: 2. Imam Arifin, ST., MT.
: 3. Mahardhika Pratama, M.Sc., Ph.D.

ABSTRACT

Deep learning is one of machine learning that enables computers to learn from data and experience. One of the algorithm in machine learning is Convolutional Neural Network (CNN) which is often used in object detection. However, there are several problem include when the object is blocked to another object, the result of accuracy is decreases and low resolution make the object not being detected. Therefore, in this research the R-CNN Faster method was used to measure the performance of object detection systems using two CNN models namely Resnet50 and VGG-16. The results of tests obtained using the ResNet50 model have an error of 0,3471 with an accuracy rate of 85,33%. While the test results with the VGG-16 model have an error of 0,6782 with an accuracy of 93,39%

Keywords : Convolutional neural network, image classification, ResNet50, VGG-16.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan rahmat dan karunianya, sehingga penulis dapat membuat dan menyelesaikan laporan tugas akhir dengan judul “Analisa Performa Sistem Deteksi Objek Konstruksi Bangunan Menggunakan *Convolutional Neural Network*”

Laporan Tugas Akhir ini bisa diselesaikan tidak terlepas dari ulur tangan berbagai pihak, oleh karna itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Segekap keluarga besar penulis khususnya kedua orang tua yang selalu mendukung dan terus menerus mendoakan sehingga dapat terselesaikanya buku Tugas Akhir ini.
2. Bapak Prof. Dr. Ir. Achmad Jazidie, M.Sc., Bapak Imam Arifin, ST., MT. dan Bapak Mahardhika Pratama M.Sc., Ph.D. yang telah membimbing penulis sehingga terselesaikannya Tugas Akhir ini.
3. Bapak Ir. Ali Fatoni, MT., Bapak Ir. Joko Susila, MT. dan Bapak Muhamad Abdul Hady, ST., MT. selaku dosen laboratorium Teknik Pengaturan AJ-104 yang telah memberikan saran dan masukan dalam pengerjaan Tugas Akhir ini.
4. Seluruh Dosen dan Karyawan departemen Teknik Elektro yang tidak bisa saya sebutkan satu per-satu.
5. Seluruh asisten laboratorium Teknik Pengaturan AJ-104 yang telah memberikan semangat dan dukungan moral kepada penulis dalam mengerjakan Tugas Akhir.
6. Seluruh anggota laboratorium Cyber Physical, Otomasi, dan Robot Industri Departemen Teknik Elektro Otomasi yang telah memberikan bantuannya selama pengerjaan Tugas Akhir.
7. Seluruh teman D’15 (penerima beasiswa PBSB angkatan 2015) yang selalu mendukung penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa masih banyak terdapat kekurangan dari buku Tugas Akhir ini, baik dari segi materi maupun penyajiannya,

mengingat masih kurangnya pengetahuan dan pengalaman penulis. Sehingga penulis mengharapkan saran dan masukan yang membangun untuk lebih baik kedepannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat digunakan sebagaimana mestinya dan dapat bermanfaat bagi pembaca.

Surabaya, 13 Januari 2020

DAFTAR ISI

ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Metodologi	3
1.6 Sistematika Penulisan.....	3
BAB II SISTEM DETEKSI OBJEK	5
2.1 <i>Computer Vision</i>	5
2.2 <i>Image Processing</i>	5
2.3 <i>Neural Network</i>	6
2.3.1 Neuron.....	7
2.3.2 <i>Arsitektur Neural Network</i>	8
2.4 <i>Machine Learning</i>	10
2.5. <i>Deep Learning</i>	11
2.6. <i>Object Detection</i>	11
2.7 <i>Convolutional Neural Network</i>	12
2.7.1 <i>Elemen – elemen dalam CNN</i>	13
a. <i>Convolution Layer</i>	13

b. <i>Pooling Layer</i>	14
c. <i>Fully Connected Layer</i>	15
d. <i>Stride</i>	15
e. <i>Zero Padding</i>	15
f. <i>Fungsi Softmax</i>	16
g. <i>Cross Entropy</i>	16
h. <i>Dropout</i>	16
i. <i>Batch Normalization</i>	17
2.7.2 <i>Optimasi</i>	18
a. <i>ADAM</i>	18
b. <i>RMSProp</i>	18
c. <i>Stochastic Gradient Descent</i>	19
2.8 <i>Faster Region-based Convolutional Neural Network</i>	19
2.8.1 <i>Region Proposal Network</i>	20
2.8.2 <i>Intersection Over Union</i>	20
2.8.3 <i>ROI Pooling</i>	21
2.8.4 <i>RPN Loss Function</i>	22
2.9 <i>Confusion Matrix</i>	22
2.10 <i>Augmentasi</i>	23
BAB III PERANCANGAN SISTEM DETEKSI OBJEK	25
3.1 <i>Data Masukan</i>	25
3.2 <i>Flowchart Algoritma Deteksi Objek</i>	27
3.3 <i>Pendeteksian Objek</i>	30
3.4 <i>Pembangunan Arsitektur</i>	30
3.4.1 <i>Faster R-CNN</i>	31

3.4.2 <i>Residual Neural network</i> 50 (ResNet50)	31
3.4.2 Model VGG-16.....	34
3.3 Pelabelan Objek	36
3.4 Optimizer.....	37
BAB IV PENGUJIAN DAN ANALISIS.....	39
4.1 Peralatan.....	39
4.2 Pengujian Praproses Data.....	39
4.3 Pengujian Performa Model Resnet50	40
4.4 Pengujian Performa Model VGG-16	44
4.5 Pengujian Optimasi	47
4.5.1 Pengujian optimasi model Resnet50.....	48
4.5.2 Pengujian Optimasi Model VGG-16	49
4.5.3 Perbandingan akurasi model ResNet50 dan VGG-16	50
BAB V PENUTUP	53
5.1 Kesimpulan.....	53
5.2 Saran.....	53
DAFTAR PUSTAKA	55
LAMPIRAN	57
RIWAYAT PENULIS	67

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 Diagram blok tahapan sistem saraf.....	6
Gambar 2. 2 Ilustrasi model jaringan saraf tiruan	7
Gambar 2. 3 <i>Single layer feedforward</i> neuron	8
Gambar 2. 4 <i>Multi layer feedforward</i> neuron.....	9
Gambar 2. 5 <i>Recurrent network</i>	10
Gambar 2. 6 Arsitektur sederhana CNN	12
Gambar 2. 7 Ilustrasi proses konvolusi pada CNN.....	13
Gambar 2. 8 Ilustrasi prinsip kerja <i>max pooling</i>	14
Gambar 2. 9 Ilustrasi <i>zero padding</i>	15
Gambar 2. 10 Ilustrasi konvolusi dengan <i>stride 1</i>	16
Gambar 2. 11 Ilustrasi proses <i>dropout</i>	17
Gambar 2. 12 Arsitektur sederhana Faster R-CNN	20
Gambar 2. 13 Ilustrasi <i>Intersection Over Unit</i>	21
Gambar 2. 14 Variasi hasil skor IoU.....	21
Gambar 3. 1 Contoh data proyek bangunan.....	25
Gambar 3. 2 Contoh pengambilan data dengan berbagai sudut	26
Gambar 3. 3 Diagram alir sistem utama deteksi objek	27
Gambar 3. 4 Diagram alir praproses	29
Gambar 3. 5 Ilustrasi alur pendeteksian objek	30
Gambar 3. 6 Ilustrasi arsitektur Faster R-CNN.	31
Gambar 3. 7 Arsitektur model ResNet50	33
Gambar 3. 8 Ilustrasi blok residual dengan fungsi skip.....	34
Gambar 3. 9 Arsitektur model VGG-16.....	34
Gambar 3. 10 Ilustrasi perbandingan <i>bounding box</i>	39
Gambar 4. 1 Pengujian praproses data.....	39
Gambar 4. 2 Hasil total kesalahan saat 50 epoch	41
Gambar 4. 3 Hasil total kesalahan saat 80 epoch	42
Gambar 4. 4 Hasil total kesalahan saat 100 epoch	42
Gambar 4. 5 Hasil total kesalahan saat 120 epoch	43
Gambar 4. 6 Hasil total kesalahan saat 150 epoch	43
Gambar 4. 7 Hasil total kesalahan saat 50 epoch	45
Gambar 4. 8 Hasil total kesalahan saat 80 epoch	45
Gambar 4. 9 Hasil total kesalahan saat 100 epoch	46
Gambar 4. 10 Hasil total kesalahan saat 120 epoch.....	46
Gambar 4. 11 Hasil total kesalahan saat 150 epoch.....	47

Gambar 4. 12 Grafik optimasi model ResNet50	49
Gambar 4. 13 Grafik optimasi model VGG-16	50
Gambar 4. 14 Grafik perbandingan akurasi model ResNet50 dan VGG-16.....	51

DAFTAR TABEL

Tabel 2. 1 <i>Confusion Matrix</i>	22
Tabel 3.1 Spesifikasi dataset deteksi objek.....	26
Tabel 3.2 Susunan layer model ResNet50.....	32
Tabel 3.3 Susunan layer model VGG-16.....	35
Tabel 4. 1 Performa sistem model ResNet50.....	40
Tabel 4. 2 Performa sistem model VGG-16.....	44
Tabel 4. 3 Parameter awal arsitektur CNN.....	48
Tabel 4. 4 Perbandingan Optimasi model Resnet50.....	48
Tabel 4. 5 Perbandingan Optimasi model VGG-16.....	49

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Pada Bab I dijelaskan mengenai latar belakang dari penelitian yang telah dilakukan dan diawali dengan melakukan studi literatur, perancangan sistem, pengujian, dan analisis data. Selain itu, dijelaskan juga mengenai perumusan masalah, batasan masalah, tujuan, sistematika, serta metodologi yang digunakan dalam penelitian yang telah dilakukan.

1.1 Latar Belakang

Machine learning dan teori pengenalan pola telah berkembang pesat dan signifikan. Beberapa penemuan dan implementasi terkait *machine learning* memungkinkan komputer untuk belajar dari suatu data dan memprediksi pola yang mungkin terjadi sehingga dapat membantu pekerjaan manusia dalam kegiatan sehari-hari. Beberapa pengaplikasiannya dalam kehidupan sehari-hari meliputi *fraud detection*, *image classification*, *information retrieval*, dan *medical diagnosis* [1].

Salah satu pengaplikasian *machine learning* yang populer adalah klasifikasi gambar dimana sebuah gambar mengkategorikan piksel-piksel dari banyak kelas menjadi satu dari beberapa piksel yang telah diekstrak dari gambar tersebut [1]. Seiring dengan berkembangnya teknologi, berbagai bidang telah menggunakan *machine learning* untuk meningkatkan kualitas produk, seperti bidang bisnis, finansial, kesehatan, riset, dan lain-lain. Selain itu, *machine learning* mulai dikembangkan agar bekerja lebih akurat, efektif dan efisien. Dari hal tersebut muncul suatu algoritma *deep learning* yang merupakan bagian dari *machine learning*.

Convolutional Neural Network (CNN) merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data dalam dua dimensi. CNN termasuk dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan digunakan untuk mengolah data gambar. Dalam pengaplikasian klasifikasi gambar, MLP kurang sesuai karena tidak dapat menyimpan informasi spasial dari data gambar dan menganggap setiap piksel adalah fitur independen sehingga menghasilkan hasil yang kurang sesuai. Pada dasarnya CNN diterapkan untuk mengklasifikasikan gambar yang belum di olah (*raw image*) dengan meng-ekstraksi piksel gambar dan menerjemahkan dari sebuah

gambar menjadi *feature*. Dimana *feature* tersebut merepresentasikan angka-angka pada gambar. dan akan dijadikan sebagai input dalam proses klasifikasi. Dalam perkembangannya, sistem deteksi objek saat ini yang semakin canggih sebagai pengenalan objek secara real-time, dengan berbagai pendekatan yang bersifat konvensional hingga modern seperti *machine learning* banyak dikembangkan oleh para ilmuwan. Salah satu metode yang sedang populer saat ini adalah *Convolutional Neural Network*. Namun, dalam pendeteksian objek pada gambar yang mana mampu mendeteksi dengan cepat dan akurat masih terdapat beberapa kelemahan dalam proses pendeteksian. Dalam tugas akhir ini dilakukan uji performa sistem deteksi objek dengan menggunakan CNN melalui beberapa pengujian parameter untuk menentukan performa terbaik dari sistem yang dibangun.

1.2 Perumusan Masalah

Pendeteksian objek dalam sebuah gambar memegang peranan penting dalam bidang *computer vision*. Namun, terdapat beberapa kendala antara lain jika objek yang dideteksi terhalang objek lainnya hasil akurasi objek yang dihasilkan mengakibatkan penurunan dan resolusi yang rendah dapat mengakibatkan objek tidak dapat terdeteksi.

1.3 Batasan Masalah

Dalam penelitian ini permasalahan yang dibahas memiliki batasan permasalahan sebagai berikut :

1. Data gambar yang digunakan dalam penelitian diambil secara mandiri melalui kamera handphone.
2. Data yang digunakan untuk training dan pengujian dalam deteksi objek pada gambar mempunyai resolusi yang berbeda-beda.
3. Dalam implementasi klasifikasi gambar menggunakan pemrograman berbasis Python 3.

1.4 Tujuan Penelitian

Tujuan dalam penelitian ini yaitu untuk mengetahui dan menganalisis performa sistem yang dibangun dengan menggunakan beberapa parameter pendukung guna mendapatkan performa sistem yang terbaik dalam deteksi objek. Selain itu, dilakukan juga perbandingan dua model yaitu model ResNet50 dan VGG-16.

1.5 Metodologi

Pada penelitian ini dilakukan melalui beberapa tahap yaitu studi literatur untuk mencari referensi yang dapat menunjang dan pengetahuan terkait dengan deteksi objek khususnya menggunakan metode CNN. Kemudian dilakukan perancangan sistem untuk membuat model yang sesuai dimulai dengan pemodelan data pra-proses terlebih dahulu kemudian dilanjut pemodelan CNN yang bertujuan untuk mencari model yang sesuai. Selanjutnya dilakukan proses pengujian untuk menguji model CNN dan mengevaluasi untuk mengetahui seberapa baik model yang dibangun dengan menggunakan pengukuran model yakni *recall*, *precision*, dan akurasi.

1.6 Sistematika Penulisan

Pada penelitian ini disusun sistematika penulisan sesuai dengan penjelasan sebagai berikut :

BAB 1 PENDAHULUAN

Pada Bab ini dijelaskan tentang latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan, serta relevansi dari penelitian yang dilakukan.

BAB 2 SISTEM DETEKSI OBJEK

Dalam Bab ini dijelaskan mengenai teori dasar dari penelitian yang dilakukan. Teori dasar yang mendasari konsep dari klasifikasi gambar dengan menggunakan metode CNN serta beberapa library pendukung yang digunakan untuk proses *training* data pada model yang dibangun.

BAB 3 PERANCANGAN SISTEM DETEKSI OBJEK

Perancangan sistem untuk klasifikasi gambar pada Bab ini diawali dengan pra-proses data terlebih dahulu untuk dapat dilakukan pemodelan *convolutional neural network* selanjutnya dilakukan proses uji data dengan data sebenarnya.

BAB 4 PENGUJIAN DAN ANALISIS

Dalam pengujian dan analisis data akan dilakukan proses implementasi untuk pengujian model pada CNN dan pengujian data serta membandingkan keluaran yang dihasilkan antara data hasil pelatihan dan data yang sebenarnya.

BAB 5 KESIMPULAN

Dalam Bab ini akan dijelaskan suatu kesimpulan terkait hasil yang telah dicapai pada penelitian yang telah dilakukan.

BAB II

SISTEM DETEKSI OBJEK

Dalam Bab ini dijelaskan mengenai teori dasar untuk membangun algoritma CNN khususnya dalam klasifikasi gambar. Arsitektur dan beberapa teori pendukung lainnya dijelaskan dengan tujuan untuk memberikan gambaran umum mengenai deteksi objek pada gambar.

2.1 Computer Vision

Computer vision merupakan salah satu bidang keilmuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati seperti gambar atau video. Pada komputer, hal ini dilakukan dengan melakukan penangkapan gambar atau video melalui kamera, lalu dilakukan proses analisis terhadap gambar tersebut. Hasil analisis digunakan untuk melakukan keputusan-keputusan yang dibuat berdasarkan kondisi gambar atau video yang ditangkap oleh kamera. Pada dasarnya *computer vision* adalah kombinasi antara *image processing* (pengolahan gambar) dan pengenalan pola. Pengolahan gambar merupakan bidang yang berhubungan dengan proses transformasi gambar tingkat tinggi dengan tujuan untuk mendapatkan kualitas gambar yang lebih baik. Sedangkan pengenalan pola berhubungan dengan proses identifikasi objek. Dalam hal ini bertujuan untuk memperoleh informasi yang dihasilkan oleh gambar. *Computer vision* saat ini sering digunakan untuk mendeteksi wajah pada gambar (*face detection*), mengenali ekspresi wajah (*facial expression recognition*), dan pengenalan pola lainnya. Dalam implementasinya *computer vision* sering digunakan bersamaan dengan jaringan syaraf tiruan.

2.2. Image Processing

Dalam bidang *image processing* terdapat tiga bidang studi yang berkaitan dengan gambar digital, yaitu :

a. Grafika Komputer

Dalam bidang Grafika Komputer bertujuan menghasilkan gambar dengan primitif-primitif geometri seperti garis, lingkaran dan sebagainya. Primitif-primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar. Contoh data deskriptif adalah koordinat titik, panjang garis, jari-jari lingkaran, tebal

garis, warna, dan sebagainya. Grafika komputer memainkan peranan penting dalam visualisasi dan *Virtual Reality* (VR).

b. Pengolahan Gambar

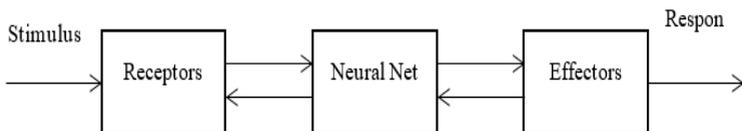
Pengolahan Gambar mempunyai tujuan untuk memperbaiki kualitas gambar agar mudah diinterpretasi oleh manusia atau mesin (komputer). Teknik-teknik dalam pengolahan gambar dengan cara mentransformasikan gambar menjadi gambar lain. Jadi, masukannya adalah gambar dan keluarannya juga gambar, namun dengan kualitas yang lebih baik daripada gambar masukan.

c. Pengenalan Pola

Pengenalan Pola mengelompokkan data numerik dan simbolik (termasuk gambar) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam gambar. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya.

2.3 Neural Network

Kecerdasan buatan merupakan salah satu bidang keilmuan yang mengembangkan suatu algoritma yang digunakan mesin atau komputer dalam menyelesaikan tugas seperti yang dilakukan oleh otak manusia. Dalam sistem jaringan saraf otak manusia terdapat kemampuan untuk membangun dan belajar dari pengalaman yang dialaminya. *Neural Network* (NN) merupakan sistem proses informasi yang memiliki karakteristik kinerja seperti jaringan saraf biologis manusia. Sistem jaringan saraf pada manusia terbagi atas tiga tahap seperti pada Gambar 2.1.



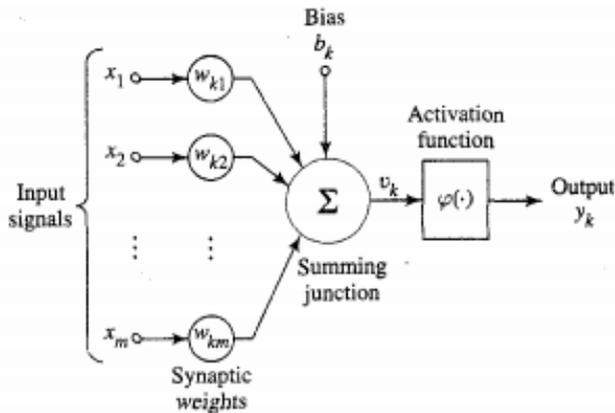
Gambar 2. 1 Diagram blok tahapan sistem saraf

Reseptor berfungsi untuk mengubah stimulus dari rangsangan yang membawa informasi untuk diproses menuju ke *neural net*. Sedangkan, *effector* berfungsi untuk merubah informasi dari *neural net* dan dijadikan sebagai keluaran. Permodelan sistem jaringan syaraf pada manusia dapat

diilustrasikan seperti pada Gambar 2.2. Dalam hal ini diadaptasi oleh *Neural Network* untuk menyelesaikan tugas tertentu.

2.3.1 Neuron

Unit pemrosesan dalam *Neural Network* disebut sebagai neuron atau *node*. Dengan kata lain, suatu sistem proses komputasional yang terinspirasi oleh bagaimana jaringan syaraf pada manusia bekerja. Dalam implementasinya neuron dapat direpresentasikan sebagai *node* yang menghubungkan antar jaringan satu dengan lainnya. Jaringan syaraf tiruan pada manusia jika dimodelkan terdapat 3 elemen dasar seperti pada Gambar 2.2



Gambar 2. 2 Ilustrasi model jaringan syaraf tiruan

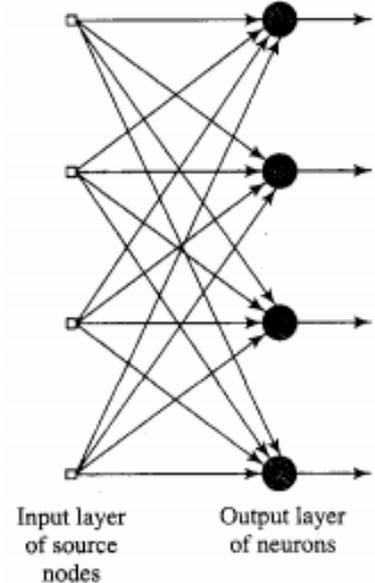
- a. Sinapsis
Sinapsis merupakan elemen yang terdapat nilai pembobotan dan sinyal masukan. Nilai pembobotan merupakan nilai random dengan rentang nilai negatif hingga nilai positif.
- b. *Summing junction*
Pada summing junction berfungsi sebagai penjumlahan untuk masing-masing nilai pembobotan sebelumnya.
- c. Fungsi aktivasi
Nilai keluaran pada neuron diperoleh dari fungsi aktivasi. Elemen ini digunakan untuk membatasi rentang sinyal keluaran pada neuron.

2.3.2 Arsitektur *Neural Network*

Pada dasarnya neuron pada neural network saling terhubung satu dengan lainnya agar dapat digunakan untuk melakukan proses pembelajaran. Secara umum arsitektur *Neural Network* dibagi menjadi 3 bagian yaitu :

a. *Single layer feedforward*

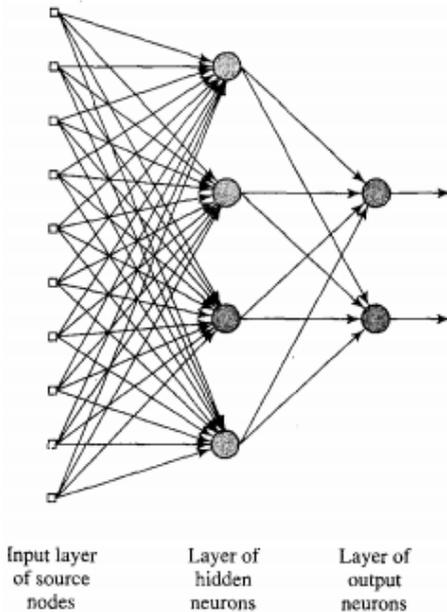
Arsitektur jaringan *single layer* dalam *neural network* terdiri dari dua layer yaitu input layer dan output layer yang memiliki beberapa neuron. Pada Gambar 2.3 diilustrasikan terdapat empat *node* masukan dalam input layer yang terhubung dengan output layer. Skema gambar tersebut merupakan *single layer network* dengan perancangan *single layer* yang didasarkan perhitungan *node* pada output layer, sedangkan untuk input layer tidak terdapat *node* perhitungan.



Gambar 2. 3 *Single layer feedforward* neuron

b. *Multi layer feedforward*

Pada jaringan *multi layer feedforward* dibedakan dengan adanya satu atau lebih *hidden layer* yang mana *node* perhitungannya disebut sebagai *hidden neuron* atau *hidden unit*. Fungsi dari *hidden neuron* yaitu untuk melakukan proses perhitungan dengan orde tinggi. Pada Gambar 2.4 menunjukkan *fully connected feedforward* dimana tiap *node* di setiap layer terhubung ke tiap-tiap *node* yang lain pada layer berikutnya. Namun, jika terdapat salah satu *node* yang tidak terhubung disebut sebagai *partially connected* [2].

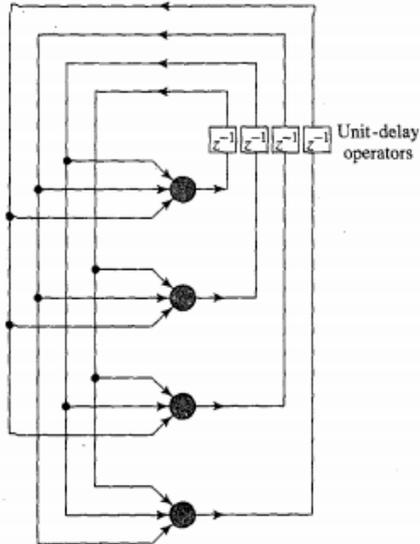


Gambar 2. 4 *Multi layer feedforward* neuron

c. *Recurrent Network*

Recurrent network merupakan tipe jaringan yang berbeda dengan lainnya. Dalam *recurrent network* mempunyai satu atau lebih *feedback loop*. Pada Gambar 2.5 menunjukkan ilustrasi *recurrent network* dimana keluaran tiap neuron dijadikan sebagai masukan dari neuron lainnya.

Tujuan dari jaringan *feedback* yaitu untuk meningkatkan kemampuan dalam proses *training* data.



Gambar 2. 5 Recurrent network

2.4 Machine learning

Machine learning merupakan salah satu cabang dalam *artificial intelligence* yang memungkinkan komputer dapat belajar dari data dan pengalamannya. Menurut seorang peneliti dalam *machine learning* Arthur Samuel, 1959 mengatakan bahwa: “*Machine learning* merupakan cabang ilmu yang mempelajari cara membuat komputer dapat belajar tanpa memprogramnya secara eksplisit”.

Dalam *machine learning* terdapat tiga metode dalam proses pembelajaran yaitu :

a. *Supervised*

Supervised learning merupakan metode pembelajaran dimana data yang akan digunakan sebagai pembelajaran diberi pelabelan untuk dapat

diklasifikasikan per kelas dan dijadikan sebagai model prediksi dari data tersebut.

b. *Unsupervised*

Unsupervised learning merupakan metode pembelajaran dimana data yang akan digunakan tidak diberikan pelabelan. Sehingga komputer dapat belajar dari data tersebut.

c. *Reinforcement Learning*

Reinforcement learning menerapkan metode diberlakukan reward sebagai hasil dari pembelajaran yang positif dan punishment pada hasil pembelajaran yang negatif. Komputer dilatih dengan cara belajar dari pengalaman melalui *trial and error* saat melakukan tugas tertentu. Dan memaksimalkan *reward* untuk belajar secara berulang-ulang.

2.5 Deep Learning

Deep learning merupakan salah satu cabang dari *Machine Learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data dan sangat kompleks. Sehingga, dibutuhkan suatu pemodelan untuk pembelajaran yang dibangun berdasarkan jaringan saraf tiruan. *Deep Learning* memungkinkan suatu model komputasional yang tersusun dari beberapa layer pemrosesan untuk belajar dari data dengan berbagai level abstraksi.

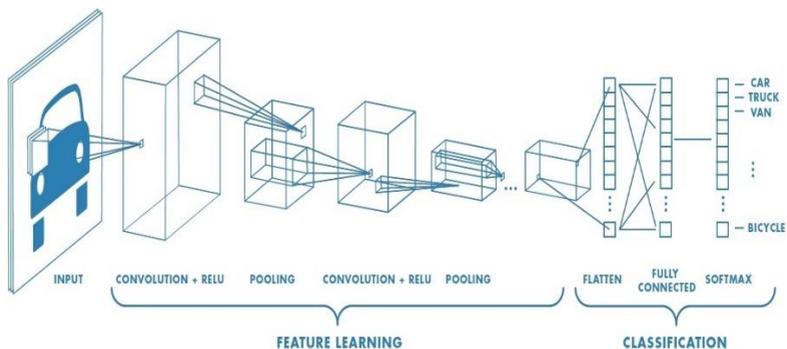
2.6. Object Detection

Perkembangan *object detection* mulai berkembang seiring dengan berkembangnya algoritma CNN yang dimulai pada tahun 1998 oleh LeCun et al. Pada tahun tersebut mereka mengembangkan suatu model CNN yaitu LeNet-5 dengan menggunakan pionier dari jaringan konvolusional 7 level yang dapat mengklasifikasikan digit dan banyak diterapkan oleh beberapa bank untuk mengenali nomor tulisan tangan pada cek yang didigitalkan dalam bentuk gambar *grayscale* berukuran 32x32 piksel. Namun pada saat itu kemampuan memproses gambar dengan resolusi yang lebih tinggi masih belum dapat dilakukan karena keterbatasan sumber daya komputasi. *ImageNet* adalah database visual besar yang dirancang untuk digunakan dalam penelitian perangkat lunak pada bidang pengenalan objek visual. *ImageNet Large Scale Visual Recognition Challenge* (IL-SVRC) merupakan ajang kompetisi untuk mengklasifikasikan dan mendeteksi objek dengan benar. Pada tahun 2012, AlexNet memenangkan ILSVRC 2012, dimana mereka menggunakan arsitektur yang mirip dengan LeNet namun dengan *layer*

yang lebih dalam, layer yang lebih banyak pada tiap *layer-nya*, dan tumpukan *convolutional layers*. AlexNet berhasil memenangkan ILSVRC dengan top-5 kesalahan sebesar 15,3%. Sejak keberhasilan AlexNet, ILSVRC selalu dimenangkan oleh metode berbasis CNN, mulai dari ZFNet (2013), GoogleNet/Inception dan VGGNet sebagai *runner-up* (2014), dan ResNet (2015) yang top-5 kesalahannya berhasil mengalahkan level kemampuan manusia dengan nilai 3,57%. Namun, masih terdapat sebuah masalah, yaitu CNN masih terlalu lambat dan biaya komputasinya yang sangat mahal. R-CNN memecahkan masalah ini dengan menggunakan algoritma *object proposal* yang disebut dengan *selective search* yang bertujuan untuk mengurangi jumlah *bounding box* yang diberikan kepada *classifier* sampai sekitar 2000 *region proposal*.

2.7 Convolutional Neural Network

Convolutional neural network (CNN) merupakan salah satu metode dalam *deep learning* yang digunakan untuk pengenalan suatu pola, seperti pengolahan gambar dan pengenalan suara. Pada Gambar 2.6 menunjukkan ilustrasi dari arsitektur sederhana CNN. Arsitektur CNN terinspirasi dari bidang keilmuan biologis pada manusia melalui beberapa model dan fitur yang dapat belajar secara hirarkis dan otomatis. Prinsip kerja dari CNN yaitu mengidentifikasi suatu *feature* pada model dan komputer belajar untuk mengenalinya kemudian menggabungkan beberapa *feature* untuk belajar terhadap pola yang lebih rumit[8]. Pada dasarnya CNN tersusun dari *feature learning* dan *classification* yang terdiri dari beberapa elemen didalamnya.



Gambar 2. 6 Arsitektur sederhana CNN

2.7. 1 Elemen – elemen dalam CNN

Beberapa elemen penting dalam membangun suatu model CNN pada dasarnya terdiri dari :

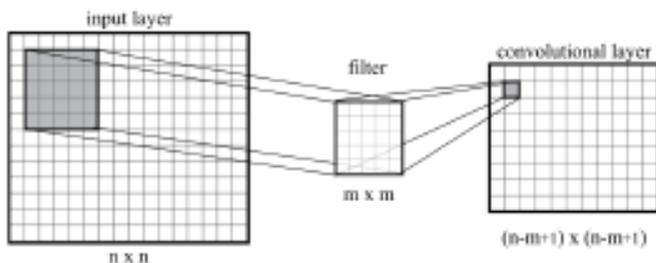
a. Convolution Layer

Convolutional layer merupakan fitur ekstraksi yang memuat ukuran volume $[W1 \times H1 \times D1]$ yang mana $W1$ adalah lebar, $H1$ adalah tinggi dan $D1$ adalah kedalaman. Keluaran dari neuron dalam *convolution layer* merupakan kalkulasi dari hasil yang diperoleh antara *weight* dan *local region* yang terhubung dengan volume masukan. Sedangkan keluaran yang diperoleh dari volume $[W2 \times H2 \times D2]$ disebut dengan *convolution maps* yang mana $W2$ merupakan lebar, $H2$ merupakan tinggi dan $D2$ merupakan kedalaman jika kita menggunakan $D2$ filter atau *convolution kernel*. *Convolution maps* menghasilkan volume yang dihasilkan berdasarkan dari persamaan (2.1), (2.2), dan (2.3).

$$w2 = \frac{w1 - F + 2 * P}{S} + 1 \quad (2.1)$$

$$H2 = \frac{H1 - F + 2 * P}{S} + 1 \quad (2.2)$$

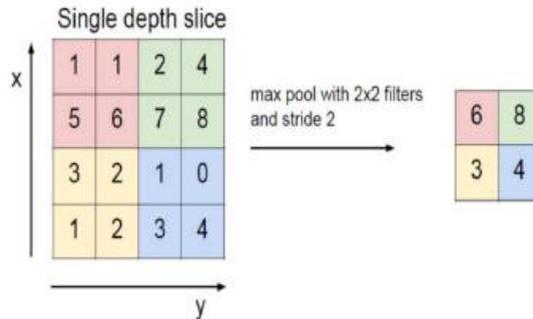
$$D2 = K \quad (2.3)$$



Gambar 2. 7 Ilustrasi proses konvolusi pada CNN

b. Pooling Layer

Fungsi utama pada pooling layer yaitu *down-sampling* untuk mengurangi kompleksitas dari beberapa layer. Dalam hal pengolahan gambar istilah *down-sampling* dapat diartikan sebagai mereduksi tingkat resolusi pada gambar. *Pooling layer* tidak mempunyai efek khusus terhadap nilai dari filter. Terdapat 2 tipe dalam *pooling layer* antara lain *max pooling* dan *average pooling*. Dalam CNN biasa digunakan metode *max pooling* yaitu membagi keluaran dari *convolution layer* dan menjadi beberapa matriks kecil kemudian diambil nilai terbesar dari tiap matrik yang telah direduksi sebelumnya. Sedangkan pada *average pooling layer* yakni mengambil nilai rata-rata pada tiap matrik yang telah direduksi. Pada Gambar 10. menunjukkan prinsip kerja dari *max pooling* yang menggunakan 2×2 filter dan stride 2 dalam arti lain tiap matrik 2×2 akan diambil nilai tertinggi dan menggesernya sebanyak 2 kotak.



Gambar 2. 8 Ilustrasi prinsip kerja *max pooling*

c. Fully Connected Layer

Pada *fully connected layer*, tiap layer dalam jaringan akan terhubung satu dengan lainnya. Dalam penerapannya *fully connected layer* sama dengan *multi layer perceptron* (MLP) dengan tujuan untuk mentransformasikan dimensi pada data agar dapat diklasifikasikan secara linear. Data masukan dari *convolution layer* sebelumnya ditransformasikan terlebih dahulu menjadi data satu dimensi atau disebut sebagai *feature vector* sebelum dijadikan masukan dari *fully connected layer* agar tidak kehilangan informasi spasialnya

d. Zero padding

Dalam CNN terdapat teknik yang digunakan untuk memanipulasi dimensi pada masukan gambar untuk penambahan nilai 0 disetiap tepi gambar. Dikarenakan metode konvolusi mempunyai kelemahan yakni hilangnya informasi pada gambar terutama pada sisi tepi gambar. Maka dari itu adanya metode *padding* / *zero padding* digunakan untuk menyelesaikan permasalahan tersebut. Dengan adanya metode tersebut dimensi keluaran akan sama atau tidak berubah secara signifikan seperti dimensi masukan. Manfaat dari penggunaan *zero padding* dapat dilihat pada Gambar 2.9. dapat diasumsikan $N = 7$, $F = 3$, dan $\text{Stride} = 1$ maka keluaran akan menjadi 5×5 namun karna adanya penambahan satu *zero padding* maka keluaran akan tetap berukuran 7×7 yang sama seperti aktualnya.

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

Gambar 2. 9 Ilustrasi *zero padding*

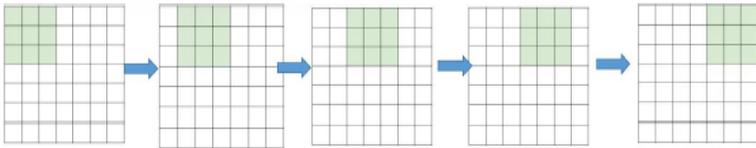
e. Stride

Dalam pengaplikasian, CNN memiliki beberapa opsi untuk menurunkan parameter-parameter dan mengurangi efek pada saat yang sama. Salah satunya yaitu *stride*. Diasumsikan pada Gambar 2.10 layer pada konvolusi berukuran 7×7 dengan menggunakan *stride* 1. Maka keluaran yang dihasilkan akan berukuran 5×5 atau ketika digunakan *stride*

2 maka keluaran yang dihasilkan 3x3. Jika dirumuskan pada Persamaan 3 sebagai berikut :

$$O = 1 + \frac{N - F}{S}$$

N = Ukuran masukan
 F = Ukuran filter
 S = Ukuran *Stride*



Gambar 2. 10 Ilustrasi konvolusi dengan *stride* 1

f. Fungsi *Softmax*

Fungsi *softmax* biasa digunakan dalam klasifikasi banyak kelas. *Softmax* memberikan nilai probabilitas untuk setiap label kelas, dimana jumlah seluruh probabilitas adalah 1. *Softmax* pada dasarnya adalah probabilitas eksponensial yang dinormalisasi dari nilai masukan sejumlah kelas pada model klasifikasi seperti pada Persamaan (2.4).

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (2.4)$$

Dimana y adalah nilai masukan. Operasi akan menghasilkan nilai probabilitas. Label dari data masukan akan ditentukan berdasarkan kelas dengan nilai probabilitas tertinggi.

g. *Cross Entropy*

Loss function merupakan fungsi yang menggambarkan kerugian yang dihasilkan oleh model. *Loss function* dikatakan baik, ketika menghasilkan *kesalahan* yang diharapkan paling rendah. Pada permasalahan klasifikasi banyak kelas, *cross entropy* adalah *loss function* yang biasa digunakan. *Cross entropy* akan menghitung *kesalahan* antara

nilai prediksi S dengan nilai sebenarnya T , seperti pada Persamaan (2.5). Selanjutnya, nilai *kesalahan* akhir diambil dari rata-rata hasil *cross entropy*, seperti pada Persamaan (2.6).

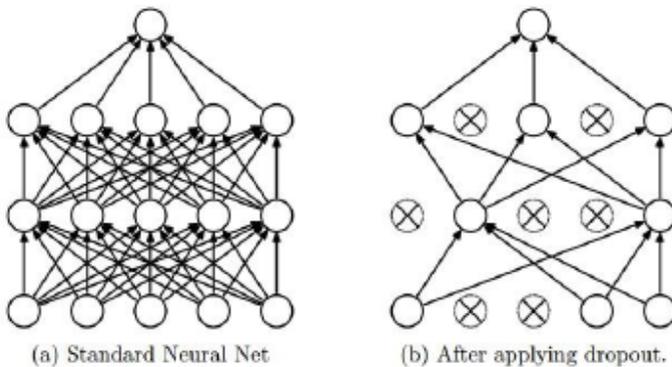
$$D(S_i, T_i) = - \sum_j T_{ij} \log S_{ij} \quad (2.5)$$

$$J(W, b) = \frac{1}{n} \sum_i D(S_i, T_i) \quad (2.6)$$

h. Dropout

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses learning. *Dropout* mengacu kepada menghilangkan neuron yang berupa *hidden layer* maupun *visible layer* di dalam jaringan. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak.

Pada Gambar 2.11 (a) neuron tetap utuh pada *neural network* yang belum memakai *Dropout*, dan (b) *neural network* yang sebagian dari neuronnya tidak digunakan setelah diaplikasikan *Dropout*.



Gambar 2. 11 Ilustrasi proses *dropout*

i. Batch Normalization

Batch Normalization adalah teknik melakukan normalisasi terhadap *batch* atau kumpulan data masukan, seperti yang terlihat pada Persamaan (2.7). Dimana x adalah nilai masukan, μ_b adalah *mean* dari *batch*, σ_b adalah standar deviasi dari *batch*. Normalisasi dilakukan agar data memiliki *mean* mendekati 0 dan standar deviasi mendekati 1.

$$x_{baru} = \frac{x - \mu_b}{\sigma_b} \quad (2.7)$$

2.7.2. Optimasi

Dalam *training* data digunakanlah optimasi sistem untuk meningkatkan performa dari sistem tersebut. Dalam *machine learning* terdapat beberapa tipe optimasi yang sering digunakan antara lain :

a. ADAM

Adaptive Moment Estimation (ADAM) adalah algoritma pengoptimalan yang biasa digunakan dalam *machine learning*. Hampir sama dengan Adagrad, Adam memiliki sebuah *learning rate* untuk setiap parameter dan secara terpisah beradaptasi saat proses *training*. Adam memperbarui nilai setiap parameter seperti RMSProp. Perbedaannya Adam menggunakan gradien yang telah diperhalus dan semakin mengecil seperti yang dapat dilihat pada Persamaan (2.8). Lalu gradien tersebut akan digunakan untuk memperbarui parameter, seperti yang dapat dilihat pada Persamaan (2.9) dan (2.10).

$$m_j = \beta_1 \cdot m_{j-1} + (1 - \beta_1)g_j \quad (2.8)$$

$$s_j = \beta_2 \cdot s_{j-1} + (1 - \beta_2)(g_j)^2 \quad (2.9)$$

$$\theta_{j+1} = \theta_j - \frac{\alpha}{\sqrt{s_j + \epsilon}} m_j \quad (2.10)$$

b. RMSProp

RMSProp (*Root Mean Square*) adalah metode pengoptimalan berbasis *adaptive learning rate* yang diusulkan oleh Geoffrey Hinton.

RMSProp memodifikasi Adagrad dengan mengganti akumulasi gradien menjadi rata-rata bergerak gradien yang diberi bobot secara kuadratik, seperti yang dapat dilihat pada Persamaan (2.11) dan (2.12).

$$S_j = \beta \cdot S_{j-1} + (1 - \beta)(g_j)^2 \quad (2.11)$$

$$\theta_{j+1} = \theta_j - \frac{a}{\sqrt{S_j + \epsilon}} g_j \quad (2.12)$$

c. *Stochastic Gradient Descent*

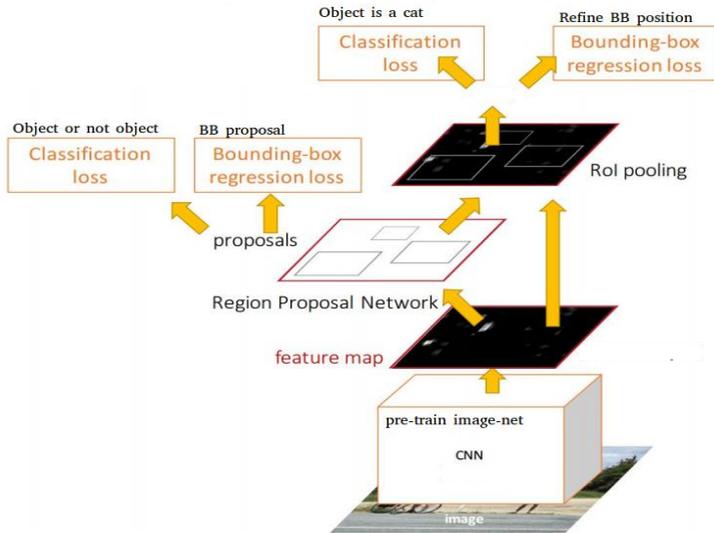
Ketika melatih sebuah model, dibutuhkan sebuah *loss function* yang dapat mengukur kualitas dari setiap bobot atau parameter tertentu. *Stochastic Gradient Descent* (SGD) adalah algoritma pengotimalan. Tujuan pengoptimalan adalah untuk menemukan parameter yang dapat meminimalkan nilai *kesalahan* dari *loss function*. SGD adalah algoritma yang digunakan untuk memperbarui nilai bobot dan bias pada neuron di *neural network*. Pada dasarnya operasi yang dilakukan hanya mengurangi bobot awal dengan sebagian nilai dari nilai gradien yang sudah kita dapat. Nilai sebagian disini diwakili oleh parameter bernama *learning rate*, seperti yang terlihat pada Persamaan (2.13) dan (2.14).

$$W_{j+1} = w_j - a \frac{\partial}{\partial w_j} J(W, b) \quad (2.13)$$

$$b_{j+1} = w_j - a \frac{\partial}{\partial b_j} J(W, b) \quad (2.14)$$

2.8 *Faster Region-based Convolutional Neural Network*

Dalam pendeteksian objek pada gambar terdapat beberapa algoritma yaitu R-CNN, Fast R-CNN, Faster R-CNN dan Mask R-CNN. *Faster Region based Convolutional Neural Network* atau bisa disebut *Faster R-CNN* merupakan algoritma yang menggunakan Fast R-CNN dan *Regional Proposal Network* (RPN) dalam arsitekturnya. Secara garis besar arsitektur Faster R-CNN dapat dilihat pada Gambar 2.12.



Gambar 2. 12 Arsitektur sederhana Faster R-CNN

2.8.1 Region Proposal Network

Regional Proposal Network biasa disebut sebagai RPN merupakan metode yang digunakan untuk menjalankan *sliding window* agar menghasilkan proposal *anchor* yang telah ditentukan. Proposal yang telah dihasilkan akan dibandingkan dengan *Ground Truth Box* menggunakan fungsi dari *Intersection Over Union* (IoU). Setiap proposal akan dilakukan proses konvolusi yang menghasilkan objektivitas objek pada gambar dengan ratio antara 0–1 yang menandakan adanya objek. Jika hasil dari IoU lebih besar dari 0,7 diasumsikan adanya objek namun sebaliknya jika hasil yang diperoleh lebih kecil dari 0,3 diasumsikan tidak adanya objek.

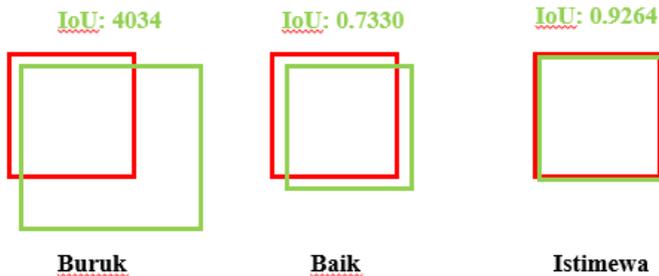
2.8.2 Intersection Over Union

Dalam menentukan letak objek terhadap suatu dataset dapat digunakan metode *Intersection Over Union* (IoU) untuk melakukan evaluasi terhadap gambar. Pada dasarnya IoU mempunyai 2 area yang

digunakan dalam menentukan akurasi terhadap objek antara lain *area ground-truth bounding box* dan area yang dideteksi dalam model yang dibangun. Dalam menentukan nilai IoU dilakukan 2 proses operasi yakni *intersection* dan *union*. Ilustrasi IoU dapat dilihat pada Gambar 2.13 dan Gambar 2.14 mengilustrasikan variasi dari hasil skor IoU yan terdiri dari 3 penilaian yakni buruk, baik, dan istimewa.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Gambar 2. 13 Ilustrasi *Intersection Over Unit*



Gambar 2 14 Variasi hasil skor IoU

2.8.3 ROI Pooling

Dalam klasifikasi gambar untuk deteksi objek pada masukan *fully connected layer* ukuran gambar harus fix. Namun, adanya kendala yang mana keluaran dari *Region Proposal Netwrk* (RPN) sangat bervariasi. *Region of Interest Pooling* (ROI) dapat dijadikan untuk merubah ukuran gambar pada RPN agar sesuai dengan masukan terhadap *fully connected layer*. ROI memanfaatkan *feature maps* untuk mengeneralisasi ukuran

dari tiap *region proposal network* sebelum dijadikan masukan sebagai *fully connected layer*.

2.8.4 RPN Loss Function

RPN *Loss Function* merupakan fungsi yang dihasilkan oleh *regional proposal network*. Dalam RPN akan menghasilkan *loss regression* dan *loss classification*. Secara matematis dapat dituliskan seperti persamaan 2.17.

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p^*_i) \lambda \frac{1}{N_{reg}} \sum_i p^*_i L_{reg}(t_i, t^*_i) \quad (2.15)$$

2.9 Confusion Matrix

Confusion Matrix merupakan sebuah tabel yang digunakan untuk mengetahui seberapa baik model yang dibangun. Dari tabel tersebut dapat dihitung tingkat akurasi, presisi, dan *recall* terhadap sistem yang dibangun.

Tabel 2. 1 Confusion Matrix

		Kelas Prediksi	
		Benar	Salah
Kelas Target	Benar	True Positive (TP)	False Negative (FN)
	Salah	False Positive (FP)	True Negative (TN)

Keterangan :

- *True Positive* (TP) menunjukkan data diklasifikasi benar (*True*) dan data aktualnya benar (*True*)
- *False Negative* (FN) menunjukkan data diklasifikasi salah (*False*) dan data aktualnya salah (*False*)
- *False Positive* (FP) menunjukkan data diklasifikasi salah (*False*) dan data aktualnya benar (*Positive*)
- *True Negatif* (TN) menunjukkan data diklasifikasi benar (*True*) dan data aktualnya salah (*False*)

Data akurasi merupakan pengukuran seberapa sistem dapat mengklasifikasikan data. Persamaan akurasi dapat dihitung melalui persamaan 2.16.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.16)$$

Presisi yaitu membandingkan jumlah data kategori positif yang dapat diklasifikasikan benar dengan data keseluruhan dengan kategori positif. Persamaan presisi dapat dilihat pada persamaan 2.17.

$$Precision = \frac{TP}{TP + FP} \quad (2.17)$$

Sedangkan *recall* yaitu pengukuran data dengan klasifikasi positif dan benar oleh sistem. Persamaan *recall* dapat dilihat pada persamaan 2.18.

$$Recall = \frac{TP}{TP + FN} \quad (2.18)$$

2.9 Augmentasi

Dalam mendapatkan performa yang optimal, umumnya *deep learning* seperti *Convolutional Neural Network* membutuhkan data yang lebih banyak dibandingkan dengan algoritma yang lain, untuk itu kita perlu melakukan augmentasi data. Augmentasi data adalah sebuah teknik menambah data dengan cara memanipulasi data yang telah ada dengan pengaturan keragaman tertentu. Untuk data berupa gambar, kita bisa lakukan operasi seperti translasi, refleksi, rotasi, perbesaran ukuran, dan lain-lain.

Halaman ini sengaja dikosongkan

BAB III

PERANCANGAN SISTEM DETEKSI OBJEK

Bab ini akan dijelaskan mengenai sistem yang akan dibangun seperti perancangan data, proses *training* menggunakan *Convolutional Neural Network* (CNN), serta gambaran umum sistem dalam bentuk diagram alir yang digunakan dalam CNN.

3.1 Data Masukan

Data masukan yang digunakan sebagai deteksi objek untuk masukan awal dalam menganalisa performa dari sistem menggunakan data gambar konstruksi bangunan. Data konstruksi bangunan diambil secara mandiri yang terdiri dari 3 kelas dengan masing-masing kelas berjumlah 130 gambar. Pada Gambar 3.1 dapat dilihat contoh konstruksi bangunan.



Gambar 3. 1 Contoh data konstruksi bangunan

Dalam proses pengambilan gambar terdapat berbagai variasi sudut pengambilan seperti tegak lurus dengan objek, miring kanan, miring kiri

dll. Pada Gambar 3.2 dapat dilihat contoh pengambilan gambar berbagai variasi sudut.



Gambar 3. 2 Contoh pengambilan data dengan berbagai sudut

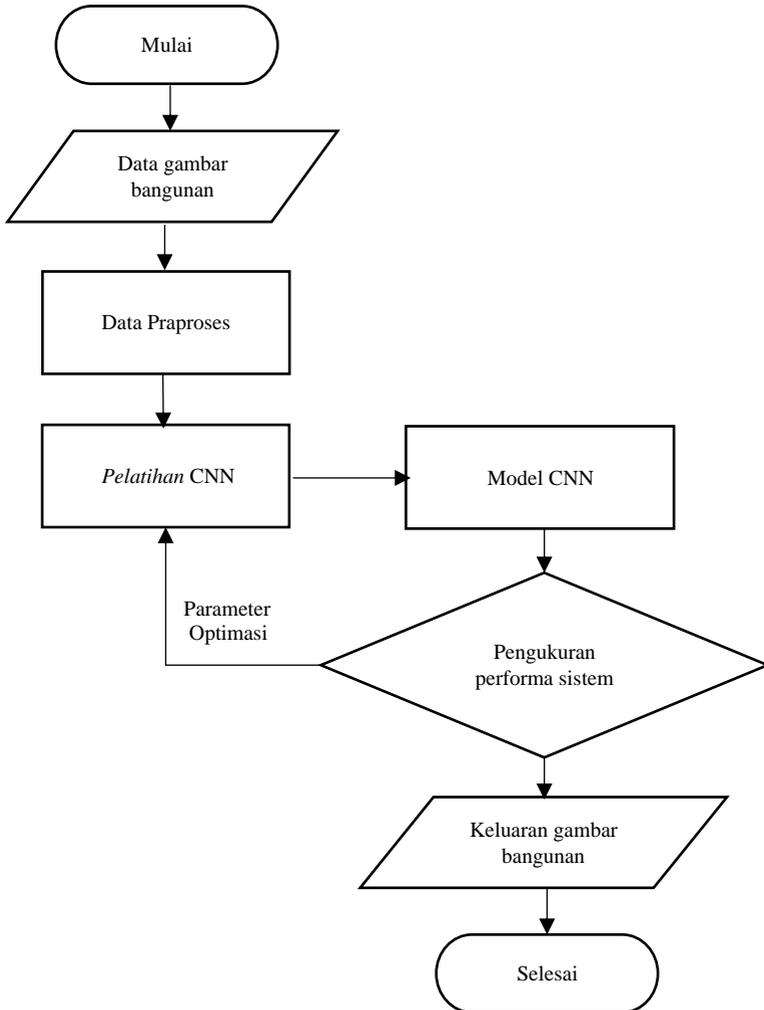
Dataset yang digunakan sebagai data masukan berjumlah 390 gambar. Dataset tersebut akan dibagi menjadi 2 bagian yaitu data *training* dan data validasi dengan pembagian data 312 gambar (80%) sebagai data *training* dan 78 gambar (20%) sebagai data validasi. Spesifikasi dataset secara lengkap dapat dilihat pada tabel 3.1.

Tabel 3.1 Spesifikasi dataset deteksi objek

Keterangan	Spesifikasi
Jenis ekstensi	.jpg
Jumlah gambar	390
Jumlah kelas	3 (Batu bata, Pintu, dan Jendela)
Jumlah gambar tiap kelas	130
Warna gambar	3 (RGB)

3.2 Flowchart Algoritma Deteksi Objek

Secara garis besar proses klasifikasi gambar menggunakan algoritma CNN didesain melalui beberapa tahap seperti terlihat pada Gambar 3.3.



Gambar 3. 3 Diagram alir sistem deteksi objek

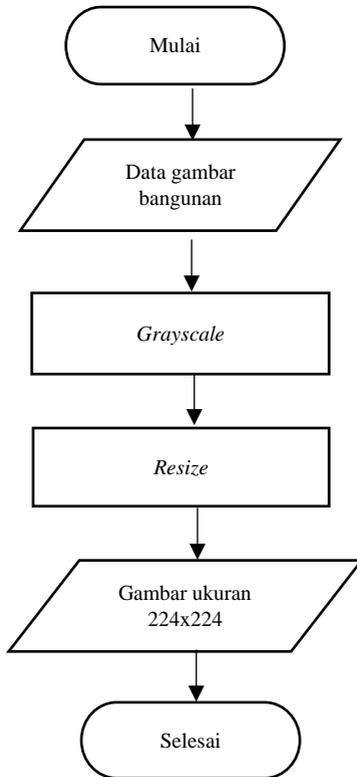
Deteksi objek pada gambar akan dilakukan tahap praproses terlebih dahulu sebelum data digunakan sebagai data *training*. Kemudian dataset akan dirubah dalam format *grayscale* sebelum dijadikan sebagai masukan proses konvolusi pada CNN dan dilakukan *resize* gambar untuk mereduksi ukuran gambar menjadi 224x224 piksel.

Pada proses *training* merupakan tahap proses untuk memperoleh model klasifikasi gambar yang terbaik dari kedua model yakni model ResNet50 dan model VGG-16 terhadap gambar konstruksi bangunan. Data *training* yang sebelumnya telah dilakukan praproses akan diekstraksi fiturnya melalui proses konvolusi, *pooling*, dan fungsi aktivasi untuk mendapatkan performa dari kedua model tersebut yang dijadikan sebagai parameter dalam membangun sistem yang diinginkan. Kemudian hasil dari proses *training* akan dijadikan sebagai model dari proses klasifikasi gambar.

Dalam proses pengujian model yang telah dibuat, dimanfaatkan fungsi *softmax* untuk mengetahui label kelas dengan dengan cara menghitung nilai probabilitas tertinggi pada suatu gambar. Hasil dari prediksi tersebut akan dijadikan sebagai acuan untuk dibandingkan dengan gambar sebenarnya. Selanjutnya dapat dilakukan evaluasi performa sistem dengan mengetahui nilai akurasi, *recall*, dan presisi pada gambar tersebut. Selain itu, dilakukan perubahan parameter optimasi untuk mengetahui optimasi terbaik dari kedua model CNN yang dibangun.

3.3 Praproses Data Pelatihan

Pada tahap praproses data, dalam hal ini gambar yang akan dijadikan sebagai masukan dari CNN akan dilakukan praproses terlebih dahulu. Praproses mempunyai tujuan agar gambar yang akan dijadikan untuk proses *training* dapat dilakukan proses konvolusi pada model CNN. Pada Gambar 3.4 menunjukkan ilustrasi tahap praproses data *training*.



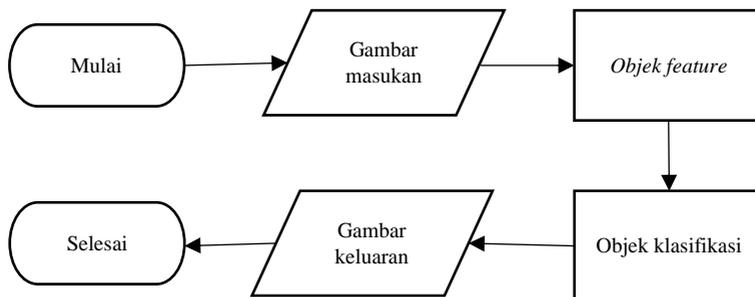
Gambar 3. 4 Diagram alir data praproses

Data masukan yang dijadikan untuk proses *training* dirubah terlebih dahulu menjadi format *grayscale* yang semula dalam format RGB. Kemudian data pada gambar akan dilakukan perubahan ukuran menjadi 224x224 piksel untuk mempercepat proses komputasi terhadap sistem yang dibangun. Selain hal tersebut dilakukan proses augmentasi untuk memperbanyak data *training* yang akan dijadikan sebagai masukan kedalam arsitektur model CNN. Selain untuk memperbanyak data *training*, proses augmentasi mempunyai tujuan untuk meningkatkan performa dan model sistem pendeteksian objek yang dibangun sesuai dengan yang diinginkan. Proses augmentasi dalam hal ini adalah

melakukan rotasi dan perbesaran ukuran data gambar konstruksi bangunan.

3.4 Pendeteksian Objek

Proses pendeteksian objek pada gambar dilakukan dengan menggunakan program python yang sebelumnya telah dibuat dan dilakukan perubahan-perubahan parameter untuk menganalisa performa terbaik dari sistem yang dibangun. Dalam penelitian ini digunakan dua model deteksi objek berbasis *convolutional neural network* yakni *Residual Neural network* (Resnet50) dan VGG-16. Dua model tersebut akan dibandingkan dengan beberapa parameter yang dibutuhkan untuk mencari performa terbaik dari sistem. Secara umum proses pendeteksian objek dapat diilustrasikan seperti pada gambar 3.5.



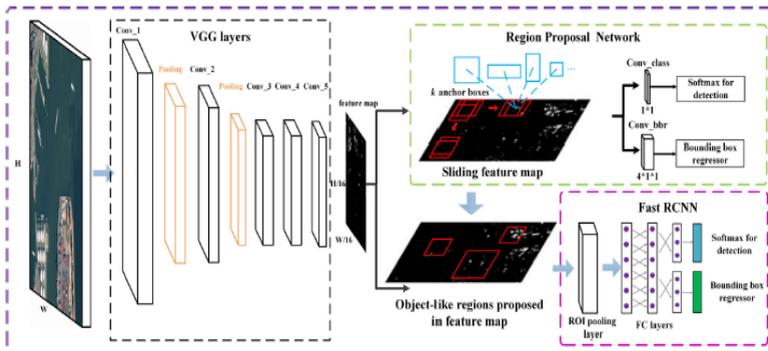
Gambar 3. 5 Ilustrasi alur pendeteksian objek

3.5 Pembangunan Arsitektur

Pembangunan arsitektur merupakan pembangunan model jaringan yang akan digunakan untuk *training* data seperti *layer* yang akan digunakan, fungsi aktivasi yang sesuai, *loss function* yang paling minimum dan beberapa parameter yang dibutuhkan dalam membangun sistem. Dalam penelitian ini digunakan dua model *convolutional neural network* yaitu model Resnet50 dan model VGG-16 yang akan dibandingkan performa sistem dari tiap masing-masing model.

3.5.1 Faster R-CNN

Faster R-CNN merupakan algoritma deteksi objek yang mana pada dasarnya terdiri dari *Fast R-CNN* dan *Regional Proposal Network* (RPN). RPN merupakan metode yang digunakan untuk menjalankan *sliding window* agar menghasilkan proposal *anchor* yang telah digunakan untuk membandingkan nilai IoU yang dihasilkan dari *ground truth bounding box* (gambar sebenarnya) dan *predicted bounding box* (gambar hasil prediksi *anchor*). *Anchor* dalam hal ini merupakan suatu area yang digunakan untuk menentukan nilai probabilitas prediksi dari gambar dan membedakan antara objek dengan *background* pada gambar. Proposal *anchor* yang digunakan Faster R-CNN mempunyai jumlah 9 *anchor* dengan 3 ratio (1:1; 1:2; 2:1) dan 3 skala (128, 256 dan 512). Pada Gambar 3.6 merupakan ilustrasi dari arsitektur Faster R-CNN dengan model VGG-16.



Gambar 3. 6 Ilustrasi arsitektur faster r-cnn.

3.5.2 Residual Neural network 50 (ResNet50)

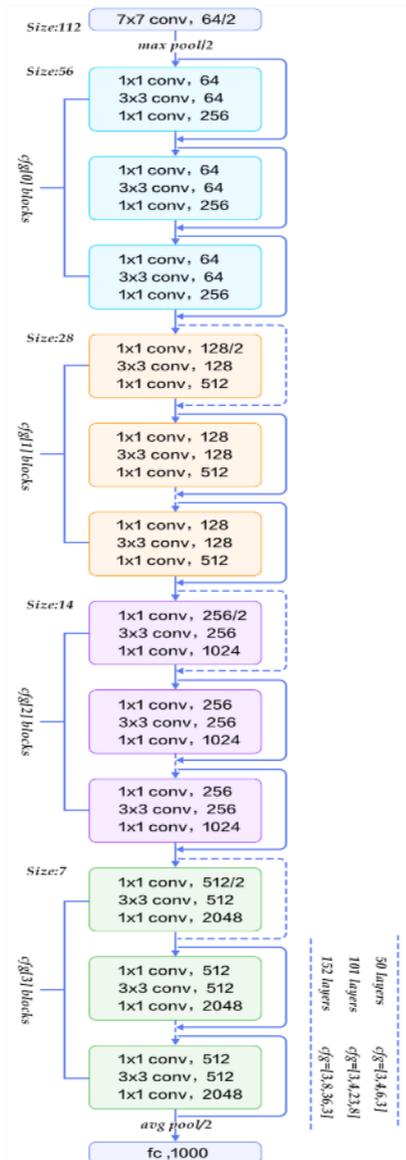
Dalam pembangunan model *Residual neural network* atau ResNet merupakan jaringan residual yang mempunyai kompleksitas yang tinggi. Pada jaringan ResNet memiliki kedalaman jaringan yang berjumlah hingga 152 lapisan. Namun, pada penelitian ini digunakan model Resnet50 yang memiliki kedalaman jaringan hingga 50 layer yang

dijadikan sebagai model CNN yang dibangun. Oleh karena itu, memungkinkan ketika dalam proses *training* menggunakan model ResNet akan menghasilkan akurasi terbaik yang dikarenakan kedalaman jaringannya. Pada Tabel 3.2 menunjukkan arsitektur dan beberapa jumlah layer yang dimiliki oleh model ResNet.

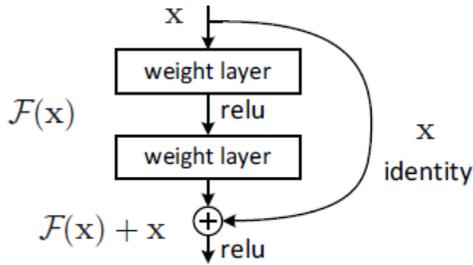
Tabel 3. 2 Susunan layer model ResNet50

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

Salah satu faktor yang disebabkan oleh hasil akurasi yang tidak sesuai atau terjadi penurunan pada jaringan untuk deteksi objek yaitu karna tidak semua jaringan dapat dioptimasi dengan mudah. Salah satu solusi yang dibuat yaitu dengan menambahkan pemetaan identitas (*identity mapping*) dengan dibuat jaringan residual atau resnet. Dalam model resnet memperkenalkan blok yang baru, disebut sebagai blok residual. Pada blok residual memiliki karakteristik yaitu jaringan saraf residual yang memanfaatkan koneksi skip atau lompatan untuk beberapa layer. Sehingga layer-layer tersebut dapat menyalin masuknya pada layer selanjutnya. Pada Gambar 3.8 merupakan ilustrasi blok residual yang menggunakan fungsi skip / lompatan. Tujuan dari lompatan terhadap beberapa layer yaitu untuk menghindari masalah terkait dengan hilangnya gradien dengan menggunakan aktivasi sebelumnya sampai dengan layer yang berdekatan.



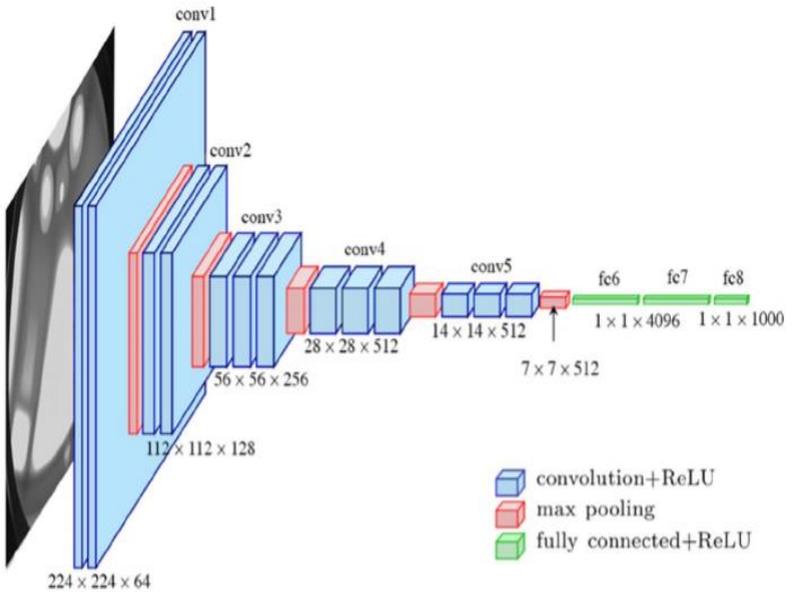
Gambar 3. 7 Arsitektur model ResNet50



Gambar 3. 8 Ilustrasi blok residual dengan fungsi skip

3.5.2 Model VGG-16

Model jaringan arsitektur VGG-16 merupakan model jaringan saraf *convolutional* yang diusulkan oleh K. Simonyan dan A. Zisserman. Pada Gambar 3.9 merupakan ilustrasi dari arsitektur model VGG-16.



Gambar 3. 9 Arsitektur model VGG-16

Model jaringan VGG-16 merupakan pengembangan / perbaikan dari model AlexNet. Pada Tabel 3.3 menunjukkan beberapa susunan layer model VGG. Pada penelitian ini digunakan model VGG-16 untuk CNN yang dibangun. Selain itu jaringan pada VGG-16 juga tersusun atas 16 layerkonvolusi dan *fully connected layer* yang digunakan untuk pengklasifikasian gambar. Arsitektur VGG-16 memiliki 13 layerkonvolusi, 5 *maxpooling* layer dan 3 *dense* layer. Dalam hal ini setiap unit *dense* mempresentasikan satu dari kategori pada gambar.

Tabel 3. 3 Susunan layer model VGG-16

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

3.6 Pelabelan Objek

Sebelum proses pendeteksian objek, dilakukan tahap proses pelabelan pada data yang akan digunakan. Proses pelabelan dilakukan untuk memberikan informasi terhadap gambar berupa kelas dan letak objek yang akan digunakan untuk deteksi. Proses pelabelan dilakukan secara manual satu-satu terhadap objek bangunan. Sehingga, akan didapatkan koordinat *ground-truth bounding box* yang dibandingkan dengan hasil dari *predicted bounding box*. Dari kedua *bounding box* tersebut akan didapatkan nilai *Intersection over Union (IoU)* untuk mengetahui letak akurasi objek pada gambar. Dalam proses pelabelan yang telah dibuat akan disimpan dalam bentuk XML dan selanjutnya disimpan dalam bentuk TXT untuk dilakukan proses *training*. Pada gambar 3.10 merupakan ilustrasi perbandingan *ground truth bounding box* dengan *predicted bounding box*.



Gambar 3. 10 Ilustrasi perbandingan *bounding box*

3.7 Optimizer

Dalam proses pelatihan dilakukan beberapa pengujian terhadap parameter optimisasi untuk memperoleh hasil yang terbaik. Dalam penelitian ini dilakukan pengujian parameter optimisasi seperti Adam *Optimizer*, Stochastic Gradient Descent (SGD) serta RMSProp.

Halaman ini sengaja dikosongka

BAB IV PENGUJIAN DAN ANALISIS DATA

Pada Bab ini akan dibahas mengenai uji coba dan analisis dari hasil perancangan sistem. Uji coba dilakukan untuk mengetahui kinerja performa dari sistem yang dibangun dengan menggunakan beberapa parameter yang telah ditentukan.

4.1 Peralatan

Dalam pengujian, perangkat yang digunakan untuk penelitian adalah laptop dengan sistem operasi windows 10 64 bit yang mempunyai spesifikasi prosesor Intel Core™ i5-5200U CPU @2.20GHz (4 CPUs), ~2.2GHz, RAM 12 GB DDR3. Sedangkan perangkat lunak yang digunakan pada pengujian sistem adalah Anaconda dengan bahasa pemrograman Python 3.6 yang dilengkapi dengan *library* Tensorflow, Keras, OpenCV, dan beberapa *library* pendukung lainnya.

4.2 Pengujian Praproses Data

Sebelum memasuki proses *training* data, dilakukan praproses gambar terlebih dahulu. Pada Gambar 4.1 menunjukkan hasil pengujian gambar pada praproses data *training*.

Gambar 4. 1 Pengujian praproses data



Pada praproses gambar terdapat beberapa proses antara lain merubah gambar yang sebelumnya mempunyai format 3 channel (*Red, Green dan Blue*) kedalam format 1 channel (*Grayscale*) dengan tujuan agar data pada gambar dapat dijadikan sebagai masukan jaringan CNN. Selanjutnya, data gambar yang telah dirubah kedalam format *grayscale* akan direduksi ukurannya menjadi 224x224 piksel dengan tujuan untuk mempercepat proses *training* dan masukan dari kedua model CNN yakni ResNet50 dan VGG-16.

4.3 Pengujian Performa Model Resnet50

Dalam pengujian data *training* menggunakan model resnet50 dengan 390 sampel data *training* dimana data dibagi menjadi 80% data sebagai data *training* dan 20% sebagai data validasi. Pada Tabel 4.1. menunjukan hasil dari performa sistem dengan beberapa variasi *epoch*.

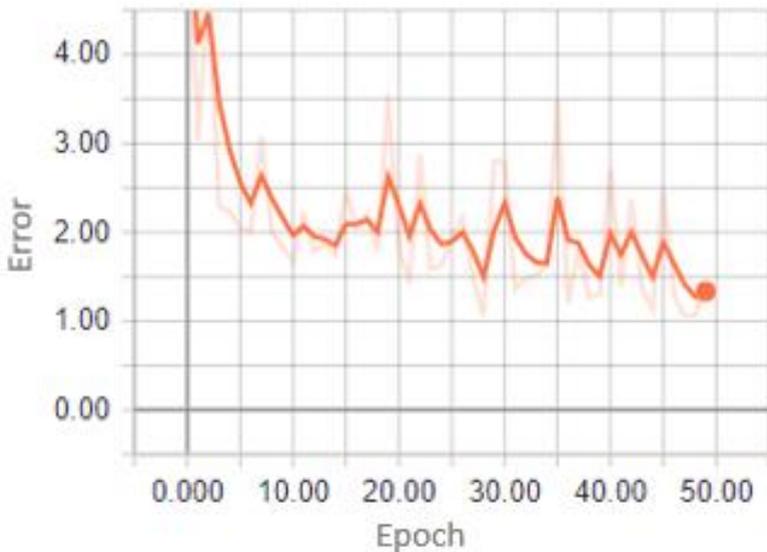
Tabel 4. 1 Performa sistem model ResNet50

Epoch	Waktu	Akurasi
50	9000 detik	41,33 %
80	16200 detik	76,00 %
100	21600 detik	77,33 %
120	25200 detik	78,66 %
150	29320 detik	85,33 %

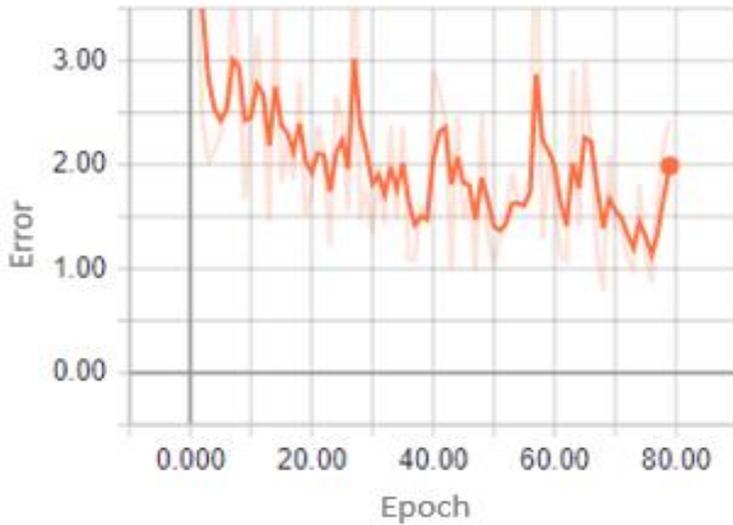
Dari hasil pengujian data dengan menggunakan model ResNet50 menghasilkan kesalahan terkecil sebesar 0,3471 dan hasil akurasi sebesar 85,33 % dalam kurun waktu untuk proses *training* selama 8 jam 14 menit. Dari tabel tersebut menunjukkan semakin besar penggunaan *epoch*. maka, hasil akurasi yang diperoleh akan semakin besar dan waktu yang dibutuhkan dalam pelatihan data akan semakin lama. Dari tabel tersebut juga menunjukan akurasi yang dihasilkan akan semakin besar. Namun, banyaknya *epoch* tidak menjamin akan menghasilkan akurasi yang tinggi. Karena dalam *training* data tidak semua gambar memiliki resolusi yang tinggi dan hal tersebut dapat berdampak pada akurasi data pengujian yang dihasilkan. Sedangkan untuk total kesalahan yang dihasilkan merupakan penjumlahan dari kesalahan RPN *Classification*, kesalahan RPN *Regression*, kesalahan *Classification*, dan kesalahan *Bounding box*

Regression. Kesalahan RPN *Classification* dan RPN *Regression* yang dihasilkan oleh *Region Proposal Network*. Sedangkan kesalahan *classification* dan *bounding box regression* merupakan kesalahan yang dihasilkan selama *training* data untuk memprediksi gambar kedalam suatu kelas dan memprediksi adanya objek dengan *bounding box* dalam gambar tersebut.

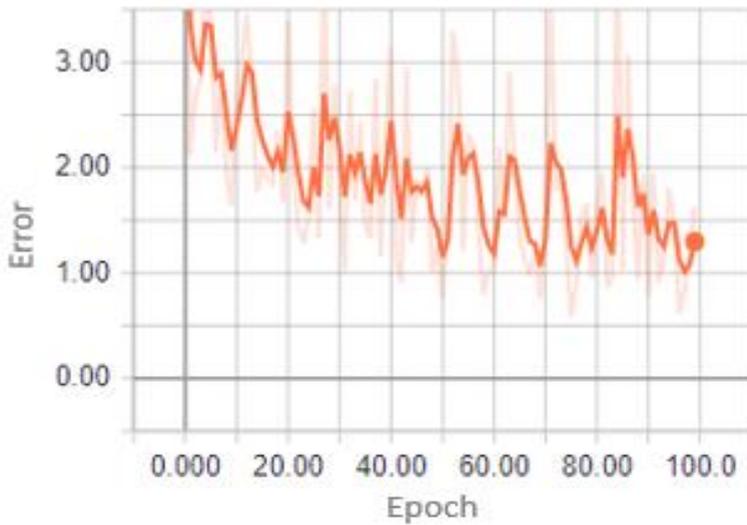
Pada Gambar 4.2, Gambar 4.3, Gambar 4.4, Gambar 4.5, Gambar 4.6 merupakan grafik total kesalahan dengan berbagai variasi mulai dari 50, 80, 100, 120, dan 150 epoch. Berdasarkan perubahan kelima grafik tersebut kesalahan terkecil diperoleh saat 150 epoch. Hal ini menunjukkan semakin banyaknya epoch yang digunakan kesalahan yang diperoleh akan semakin kecil. Namun, tidak menjamin akan menghasilkan akurasi yang tinggi.



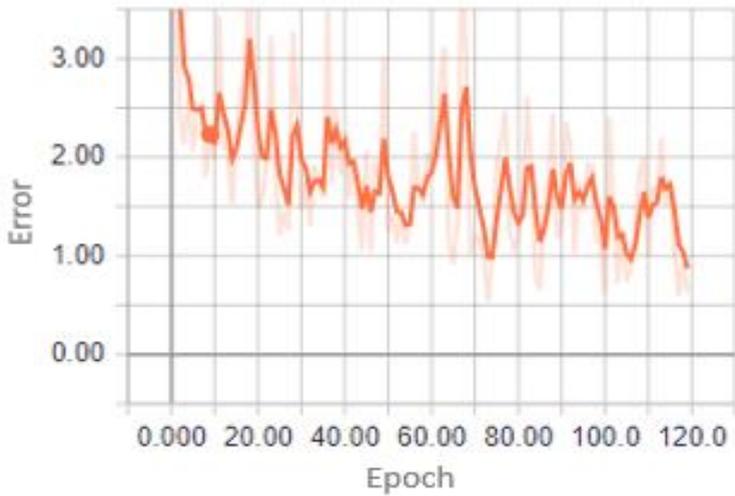
.Gambar 4. 2 Hasil total kesalahan saat 50 epoch



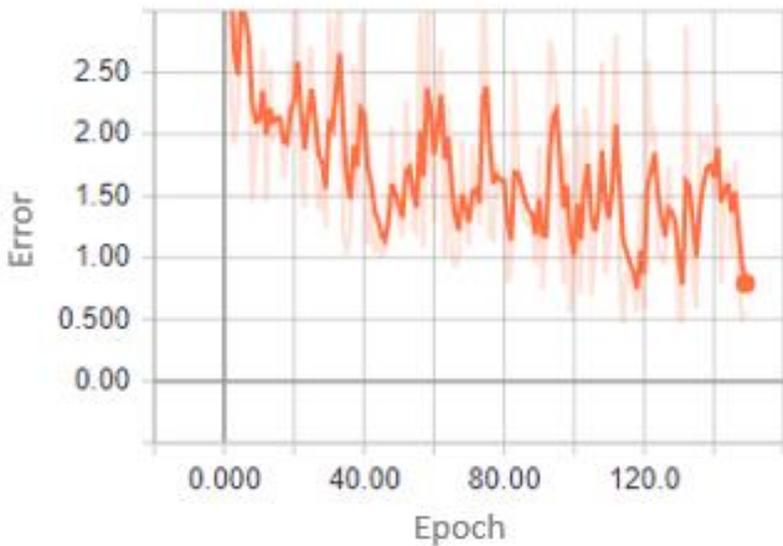
Gambar 4. 3 Hasil total kesalahan saat 80 epoch



Gambar 4. 4 Hasil total kesalahan saat 100 epoch



Gambar 4. 5 Hasil total kesalahan saat 120 epoch



Gambar 4. 6 Hasil total kesalahan saat 150 epoch

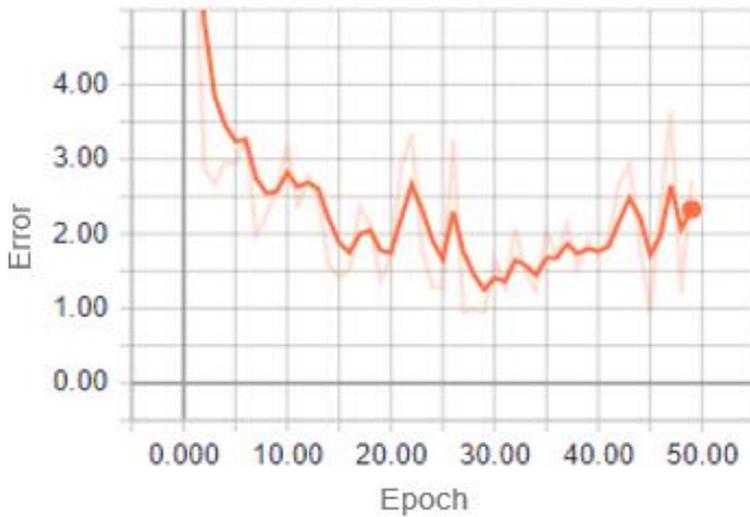
4.4 Pengujian Performa Model VGG-16

Dalam pengujian data *training* menggunakan model VGG-16 dengan 390 sampel data *training* yang mana dibagi 80% data sebagai data *training* dan 20% sebagai data validasi. Pada Tabel 4.2 menunjukkan hasil dari performa sistem dengan beberapa variasi epoch untuk mencari performa terbaik pada sistem deteksi objek menggunakan Faster R-CNN.

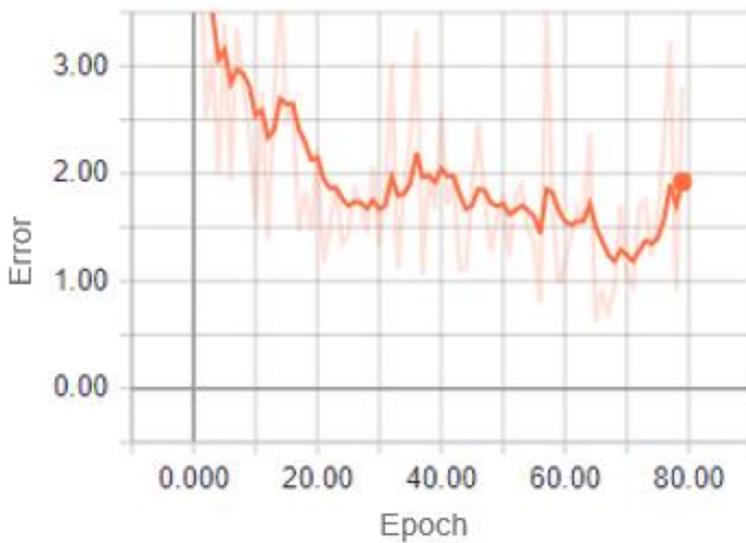
Tabel 4. 2 Performa sistem model VGG-16

Epoch	Waktu (s)	Akurasi
50	8280 detik	60,00 %
80	14680 detik	86,66 %
100	16200 detik	90,66 %
120	20880 detik	92,00 %
150	25260 detik	93,33 %

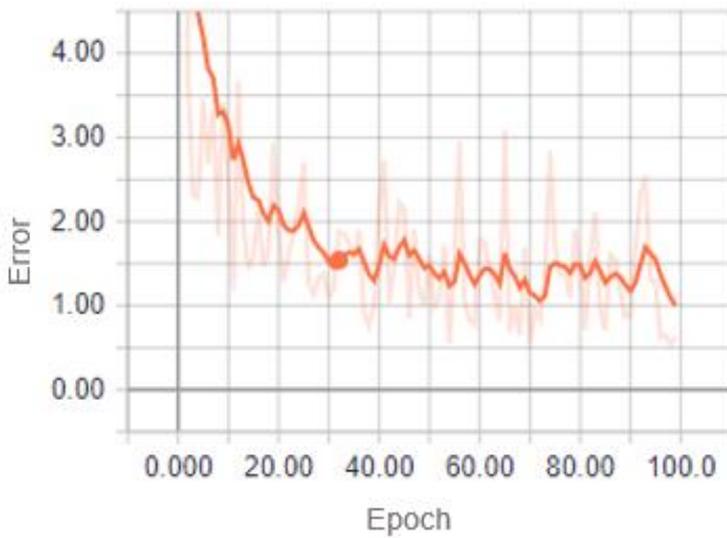
Dari hasil pengujian data dengan menggunakan model VGG-16 menghasilkan kesalahan terkecil sebesar 0,6782 dengan hasil akurasi sebesar 93,33 % dalam kurun waktu untuk proses pelatihan selama 7 jam 1 menit. Dari tabel tersebut menunjukkan semakin besar penggunaan epoch maka, hasil akurasi yang diperoleh akan semakin tinggi dan waktu yang dibutuhkan dalam pelatihan data akan semakin lama. Namun, dari pengujian data tersebut tidak dapat menjamin hasil akurasi yang didapatkan akan selalu tinggi. Karena dalam *training* data dapat disebabkan oleh beberapa faktor yang dapat mempengaruhi dari performa sistem deteksi objek antara lain gambar yang digunakan memiliki tingkat resolusi yang berbeda-beda, terhalang oleh objek lainnya, dan lain-lain. Sedangkan untuk total kesalahan yang dihasilkan merupakan penjumlahan dari kesalahan RPN *Classification*, kesalahan RPN *Regression*, kesalahan *Classification*, dan kesalahan *Bounding box Regression*.



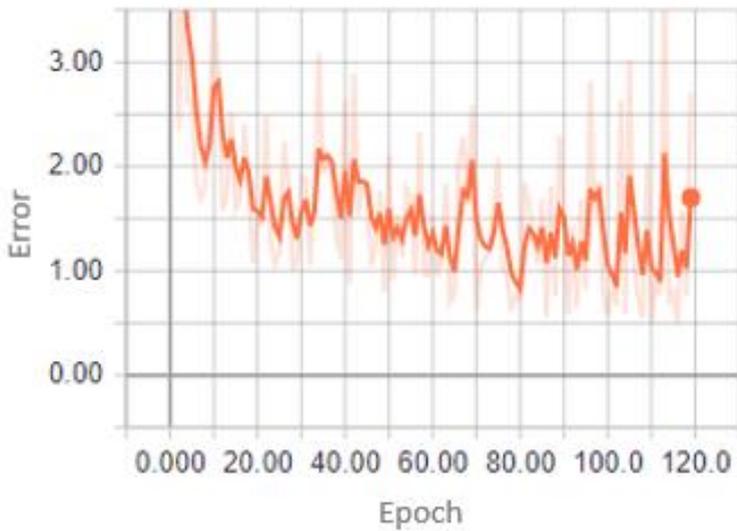
Gambar 4. 7 Hasil total kesalahan saat 50 epoch



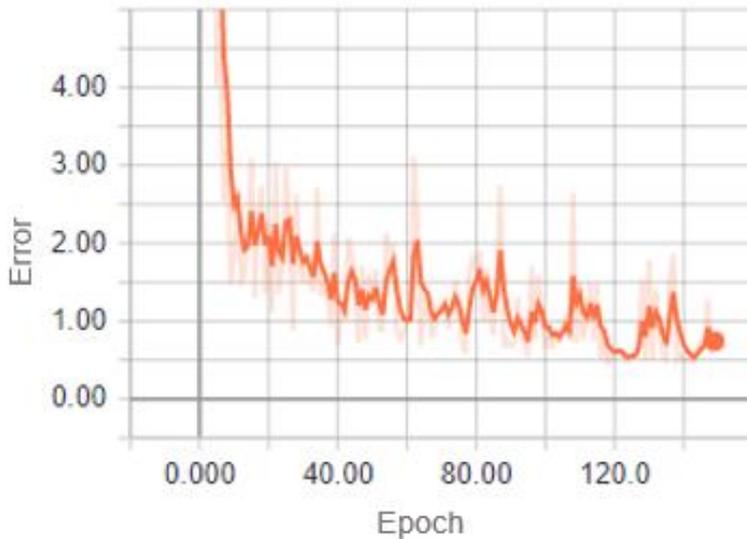
Gambar 4. 8 Hasil total kesalahan saat 80 epoch



Gambar 4. 9 Hasil total kesalahan saat 100 epoch



Gambar 4. 10 Hasil total kesalahan saat 120 epoch



Gambar 4. 11 Hasil total kesalahan saat 150 epoch

Pada Gambar 4.7, Gambar 4.8, Gambar 4.9, Gambar 4.10, Gambar 4.11 merupakan grafik total kesalahan dengan berbagai variasi mulai dari 50, 80, 100, 120, dan 150 epoch. Berdasarkan perubahan kelima grafik tersebut kesalahan terkecil diperoleh saat 150 epoch. Hal ini menunjukkan semakin banyaknya epoch yang digunakan kesalahan yang diperoleh akan semakin kecil.

4.5 Pengujian Optimasi

Pengujian parameter optimasi bertujuan untuk menentukan parameter-parameter agar menghasilkan performa model terbaik. Penentuan parameter yang tepat akan memberikan hasil terbaik saat pengujian data. Pada penelitian ini akan digunakan 3 optimasi tiap model CNN. Pada table 4.3 merupakan parameter awal yang digunakan saat *pelatihan* dengan model resnet50 dan VGG-16.

Tabel 4. 3 Parameter awal arsitektur CNN

Keterangan	Parameter
Jumlah <i>epoch</i>	100
Ukuran <i>batch</i>	15
<i>Learning rate</i>	0,0001
<i>Loss - Function</i>	<i>Categorical Cross Entropy</i>

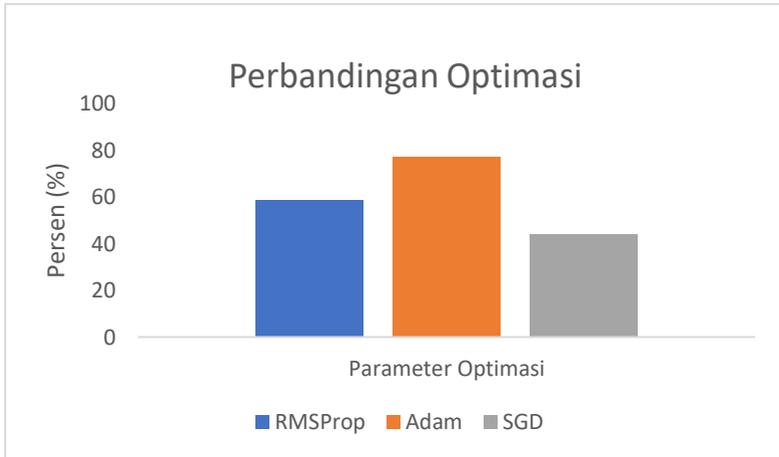
4.5.1 Pengujian optimasi model Resnet50

Pada pengujian optimasi dengan menggunakan model resnet50 digunakan 3 parameter optimasi yakni RMSProp, ADAM, dan *Stochastic Gradient Descent* (SGD) seperti pada tabel 4.4.

Tabel 4. 4 Perbandingan Optimasi model Resnet50

Optimizer	Lama waktu pelatihan	Akurasi	Loss
RMSProp	19.080 detik	58,66 %	0.830
Adam	21.600 detik	77,33 %	1.629
SGD	18.360detik	44,15 %	1.686

Dari tabel yang diperoleh hasil yang terbaik dari 3 optimasi tersebut adalah adam optimizer dengan hasil akurasi yang didapatkan sebesar 77,33 % dan kesalahan sebesar 1,629. Pada Gambar 4.12 menunjukkan perbandingan dari ketiga optimasi menggunakan model ResNet50. Dalam hal ini optimasi ADAM menunjukkan akurasi yang terbaik dibanding yang lainya karena ADAM mempunyai suatu *learning rate* untuk setiap parameter dan terpisah untuk beradaptasi saat proses *training* data. Sama halnya seperti RMSProp yang memperbarui *weight* disetiap parameternya. Namun, perbedaanya ADAM menggunakan gradient yang telah diperhalus sebelumnya. Kemudian, gradien tersebut akan digunakan untuk memperbarui parameter *weight* saat proses training data.



Gambar 4. 12 Grafik optimasi model ResNet50

4.5.2 Pengujian Optimasi Model VGG-16

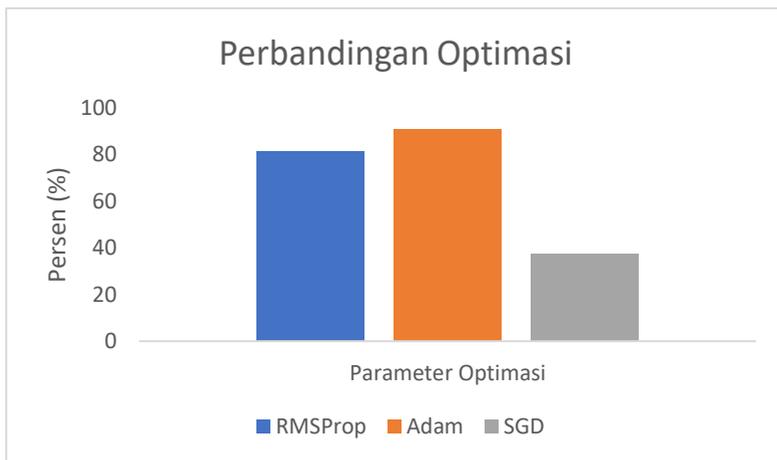
Pada pengujian optimasi dengan menggunakan model VGG-16 digunakan 3 parameter optimasi seperti pada tabel 4.5.

Tabel 4. 5 Perbandingan Optimasi model VGG-16

Optimizer	Lama waktu pelatihan	Akurasi	Loss
RMSProp	16.200 detik	81.33 %	1.7160
Adam	16.200 detik	90.66 %	0.6782
SGD	14.760 detik	37.33 %	2.1730

Dari tabel yang diperoleh hasil yang terbaik dari 3 optimisasi tersebut adalah adam optimizer dengan hasil akurasi yang didapatkan sebesar 90.66% dan kesalahan sebesar 0.6782. Pada Gambar 4.13 menunjukkan perbandingan dari ketiga optimasi menggunakan model VGG-16. Dalam hal ini optimasi ADAM menunjukkan akurasi yang terbaik dibanding

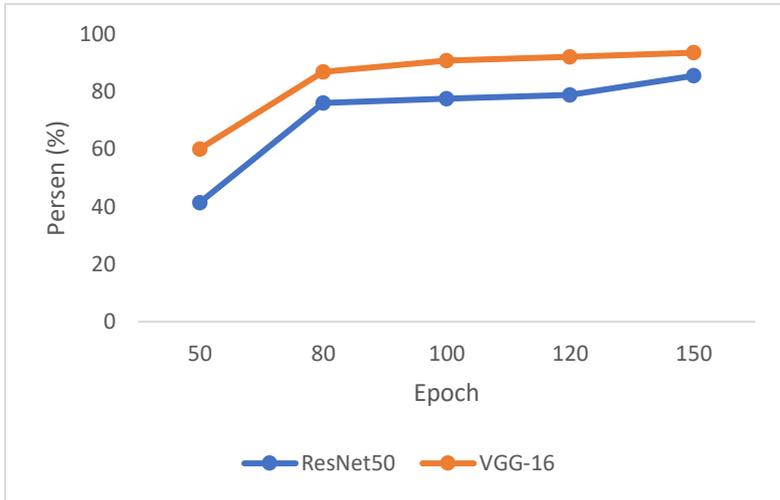
yang lainya karena ADAM mempunyai suatu *learning rate* dimana setiap parameter berbeda-beda dan terpisah untuk beradaptasi saat proses *training* data. Sama halnya seperti RMSProp yang memperbarui *weight* disetiap parameternya. Namun, perbedaanya ADAM menggunakan gradient yang telah diperhalus sebelumnya. Kemudian, gradien tersebut akan digunakan untuk memperbarui parameter *weight* saat proses training data. Selain itu, kesalahan yang didapatkan sangat kecil dibanding dengan beberapa optimasi lainnya.



Gambar 4. 13 Grafik optimasi model VGG-16

4.5.3 Perbandingan akurasi model ResNet50 dan VGG-16

Pada pengujian tingkat akurasi sistem yang dihasilkan pada model ResNet50 dan VGG-16 dapat dilihat pada Gambar 4.14. Pada grafik tersebut merupakan hasil dari pengujian dengan menggunakan epoch yang bervariasi mulai dari epoch 50, 80, 100, 120 dan 150. Akurasi tertinggi diperoleh oleh model VGG-16. Dari kedua model tersebut perbandingan optimasi terbaik diperoleh oleh model VGG-16 dengan hasil akurasi 93,33 %.



Gambar 4. 14 Grafik perbandingan akurasi model ResNet50

Halaman ini sengaja dikosongkan

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan kesalahan terkecil yang dihasilkan dengan menggunakan *Convolutional Neural Network* diperoleh ketika diberikan 150 *epoch* dengan besar tiap *epoch* 15 *batch*. Dari hasil pengujian untuk perbandingan kesalahan yang dihasilkan oleh model resnet dan model VGG16 dengan kesalahan terkecil masing-masing 0.3471 dan 0.6789 serta hasil yang diperoleh tidak menunjukkan perbedaan secara signifikan. Selanjutnya, hasil akurasi dari kedua model yakni Model ResNet50 dan VGG-16 dengan menggunakan 150 *epoch* menghasilkan akurasi sebesar 85,33% sedangkan untuk model VGG-16 sebesar 3,33%.

5.2 Saran

Berdasarkan dari penelitian yang telah dilakukan, bermula dari *epoch* yang digunakan untuk meningkatkan performa sistem deteksi objek. Perlunya pengembangan dalam hal tingkat kesalahan dan akurasi yang tinggi. Dalam penelitian ini kedepanya dapat digunakan sebagai pengembangan dari estimasi waktu pengerjaan proyek konstruksi melalui deteksi objek pada gambar.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

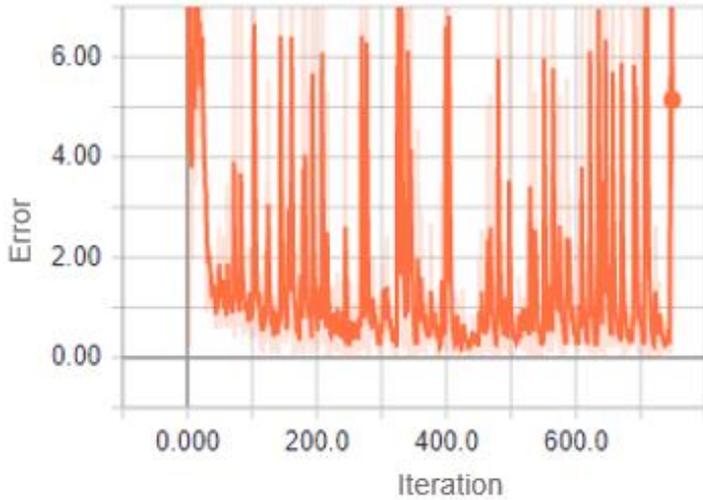
- [1] Haykin, Simon. 1931. “*Neural Network : A Comprehensive Foundation 2nd edition*”. Prentice Hall, Inc.
- [2] Fausett, Laurene. 1994. “*Fundamentals of Neural Network Architectures, Alghorithms, and Applications*”. Prentice Hall, Inc.
- [3] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.
- [4] William Travis and Li Robert “An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification” *Journal of Software and Applications* , 2018, pp. 66-88 Black & Veatch, *Power Plant Engineering*,1996.
- [5] Zhipeng Deng, et.al. “Multi Scale Object Detection in Remote Sensing Imagery with Convolutional Neural Networks” *ISPRS Journal of Photogrammetry and Remote Sensing*. 2018.
- [6] Krizhevsky, Alex et al “ImageNet Classification with Deep Convolutional Neural Networks” *Journal of Arxiv e-prints*, 2012, vol. 25.
- [7] C. Cao *et al.*, "An Improved Faster R-CNN for Small Object Detection," in *IEEE Access*, vol. 7, pp. 106838-106846, 2020..
- [6] S. Hayat, S. Kun, Z. Tengtao, Y. Yu, T. Tu and Y. Du, "A Deep Learning Framework Using Convolutional Neural Network for Multi-Class Object Recognition," *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, Chongqing, 2018, pp. 194-198.

[7] M. Shah and R. Kapdi, "Object detection using deep neural networks," *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, 2017, pp. 787-790.

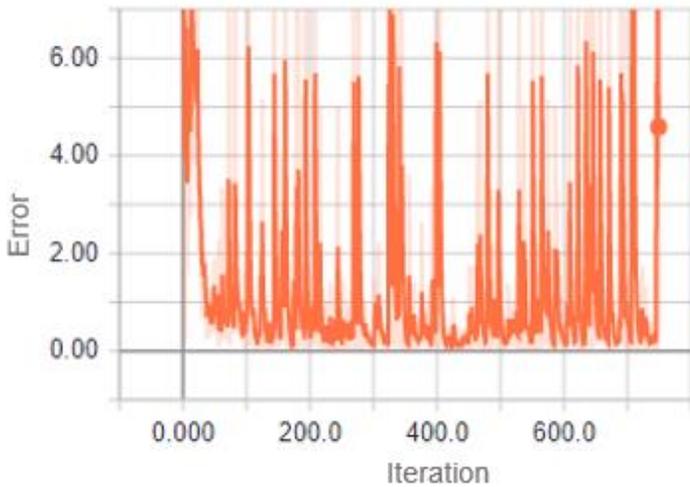
[8] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-6.

LAMPIRAN

- Pengujian 50 Epoch model VGG-16.

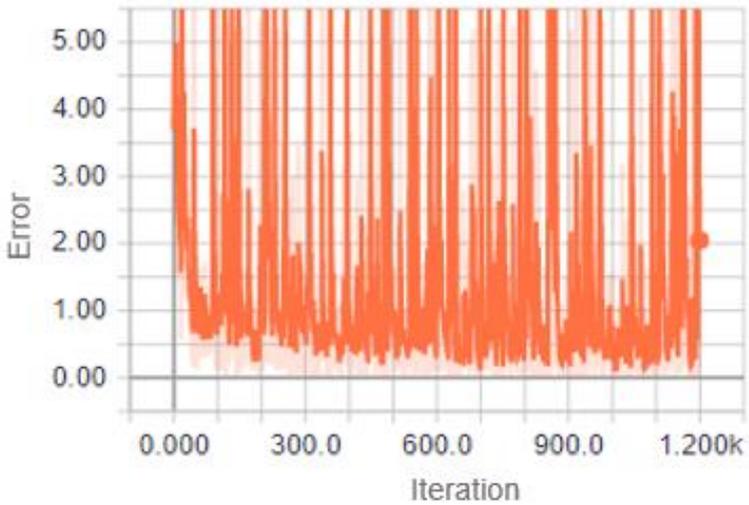


Gambar 5.1 Kesalahan RPN Classification 4,5033

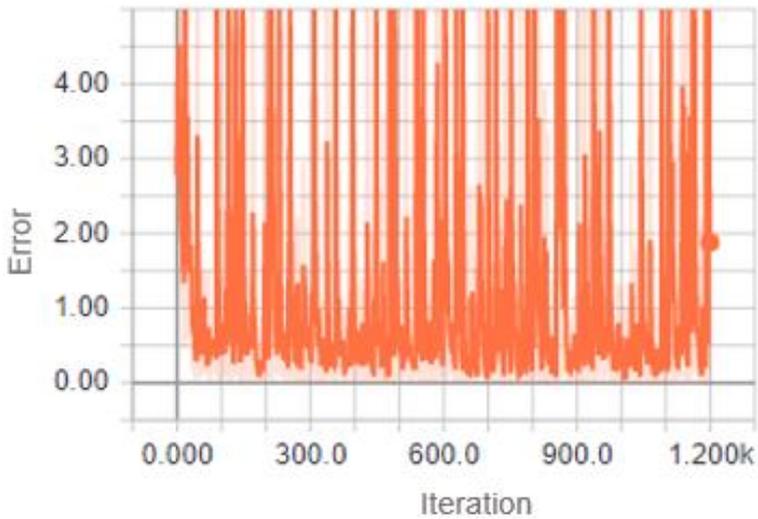


Gambar5.2. Kesalahan RPN Regression 4,3503

- **Pengujian 80 Epoch model VGG-16.**

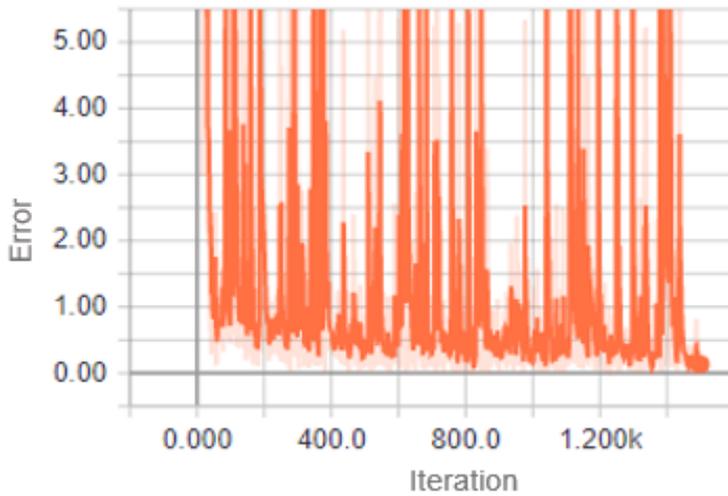


Gambar 5.3. Kesalahan RPN Classification 2,0560

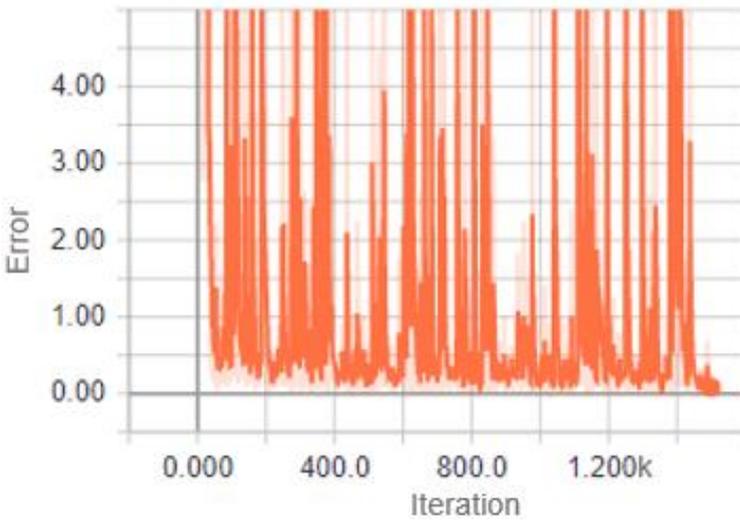


Gambar 5.4. Kesalahan RPN Regression 1,9850

- **Pengujian 100 Epoch model VGG-16.**

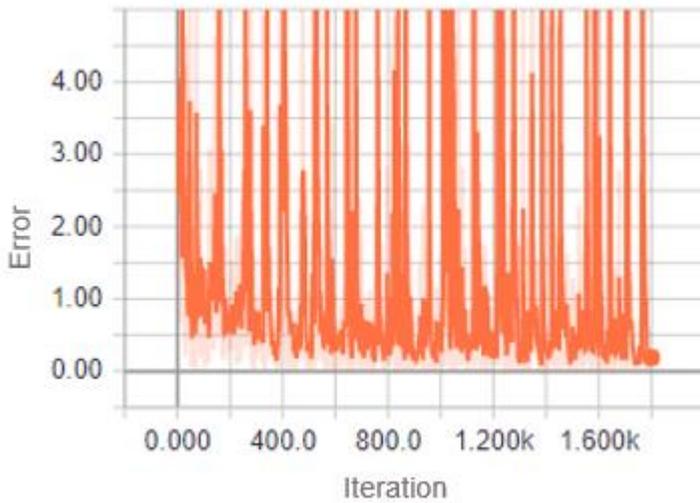


Gambar 5.5 Loss RPN Classification sebesar 0,0452

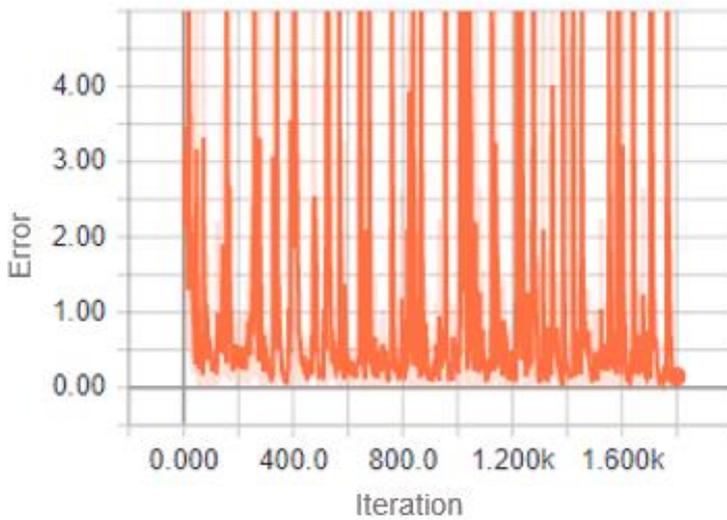


Gambar 5.6 Kesalahan RPN Regression sebesar 0,0544

- **Pengujian 120 Epoch model VGG-16.**

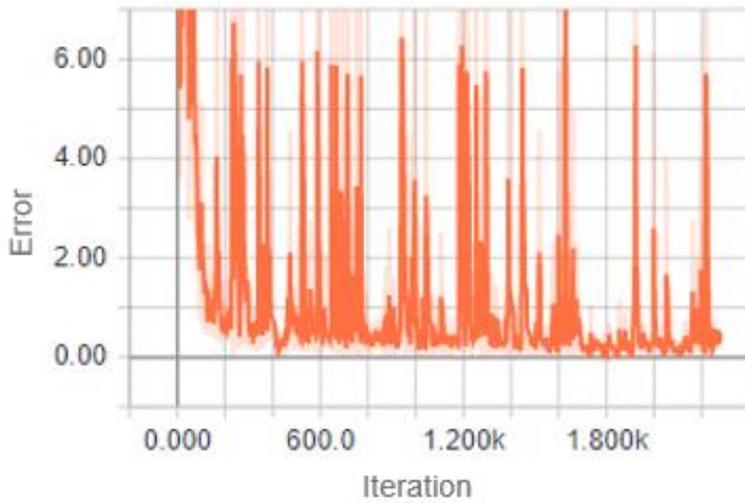


Gambar 5.7. Loss RPN Classification sebesar 0,1236

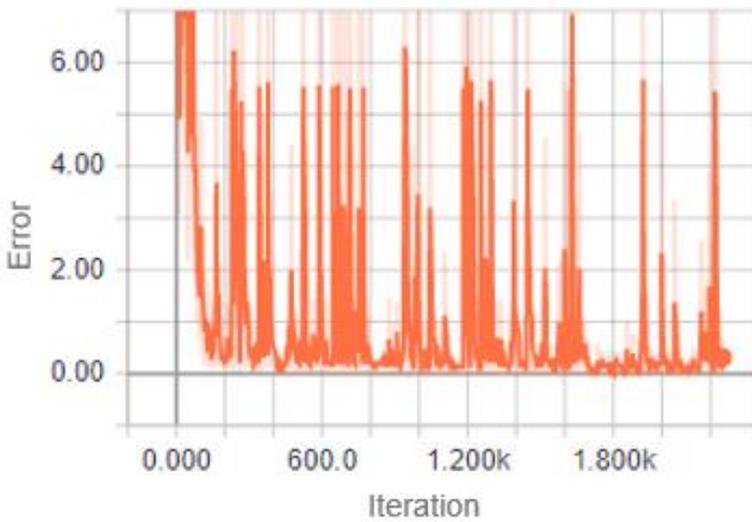


Gambar 5.8 Kesalahan RPN Regression sebesar 0,1014

- **Pengujian 150 Epoch model VGG-16.**

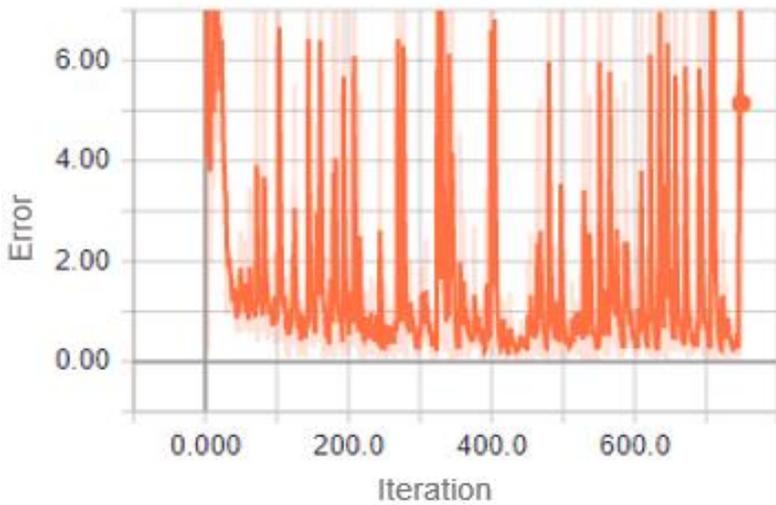


Gambar 5.9 Kesalahan RPN Classification sebesar 0,8116

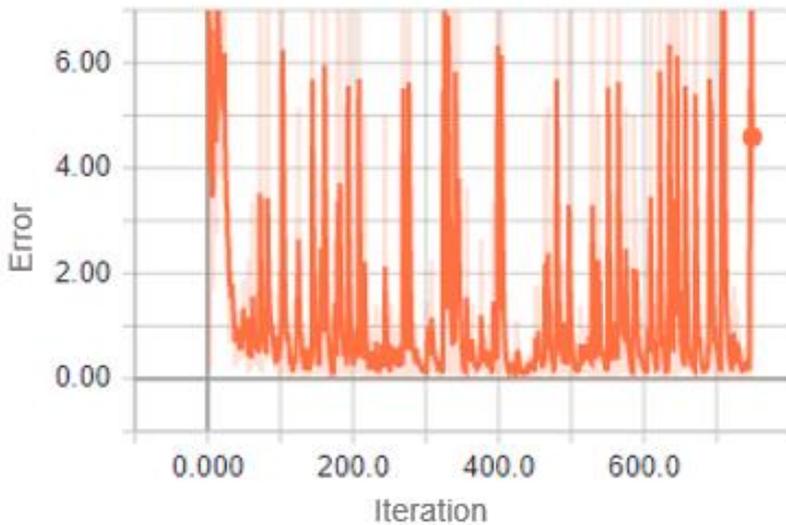


Gambar 5.10 Kesalahan RPN Classification sebesar 0,8116

- **Pengujian 50 Epoch model ResNet50.**

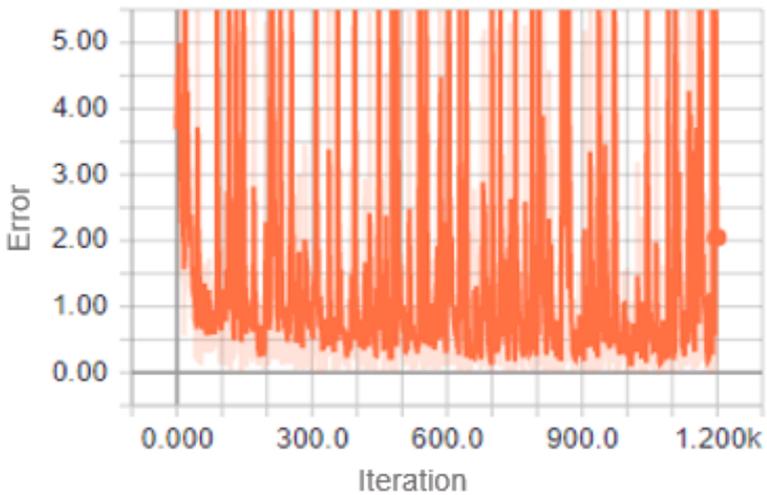


Gambar 5.11. Kesalahan RPN Classification sebesar 0,4560

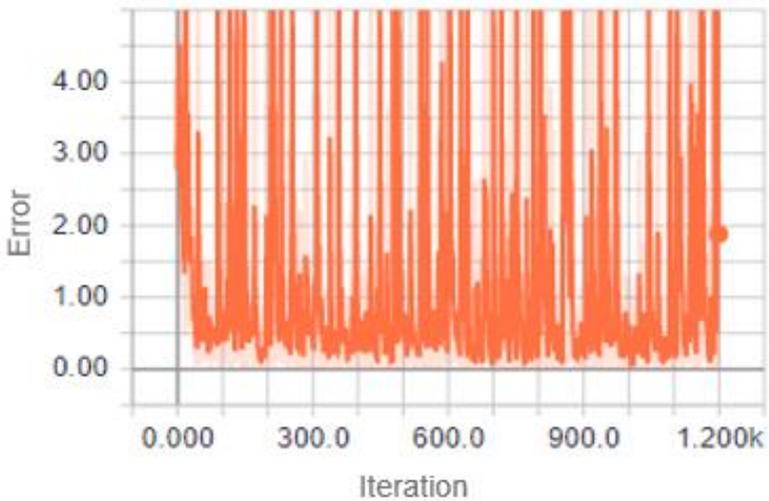


Gambar 5.12. Kesalahan RPN Regression sebesar 0,4560

- **Pengujian 80 Epoch model ResNet50.**

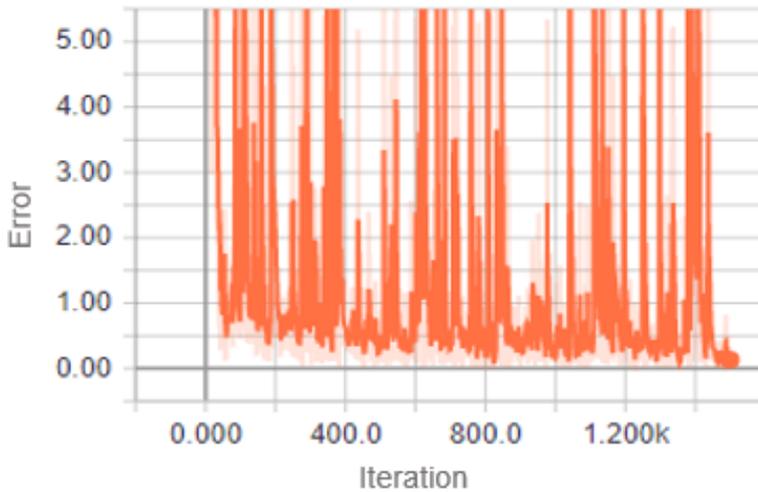


Gambar 5.13 Loss RPN Classification sebesar 0.1463

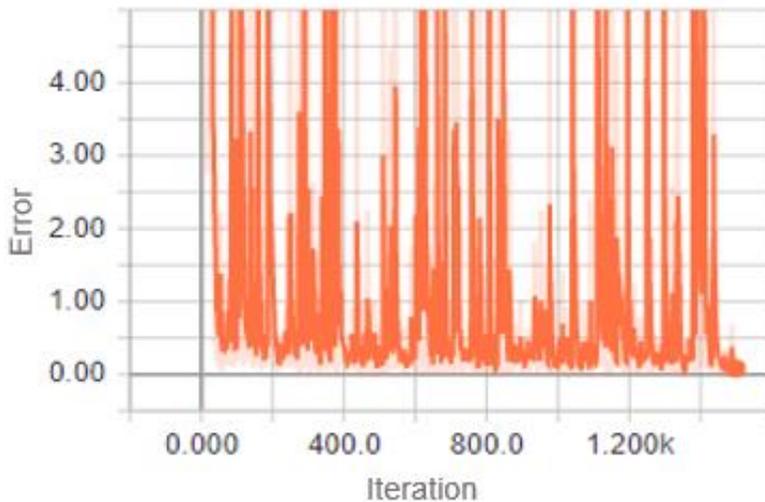


Gambar 5.14 Loss RPN Regression sebesar 0.1014

- **Pengujian 100 Epoch model ResNet50.**

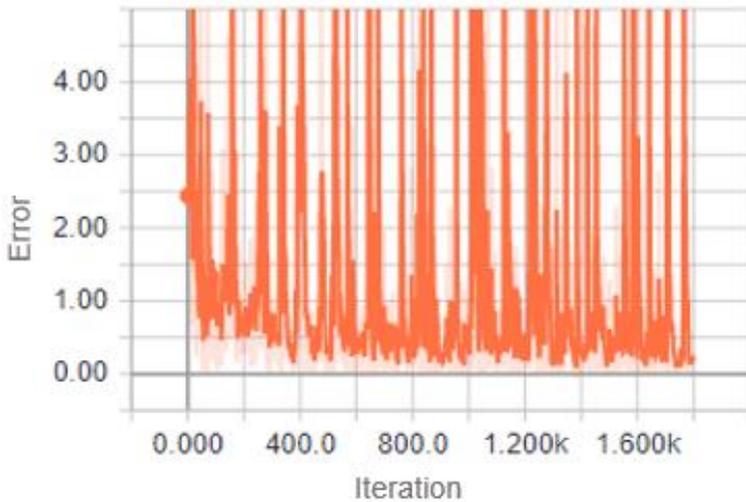


Gambar 5.15 Loss RPN Classification sebesar 0.0452

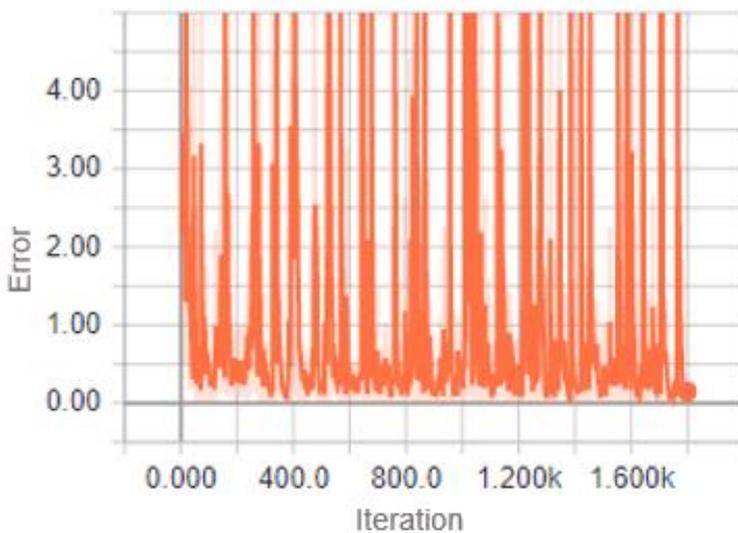


Gambar 5.16 Loss RPN Regression sebesar 0.0552

- Pengujian 120 Epoch model ResNet50.

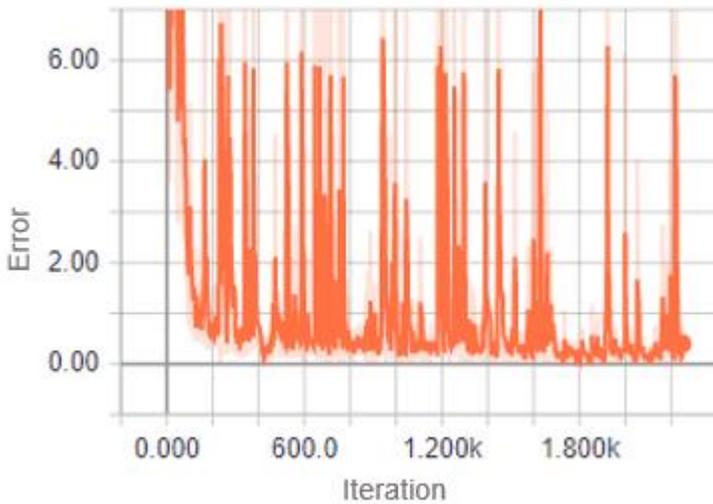


Gambar 5.17 Kesalahan RPN Classification sebesar 0.1236

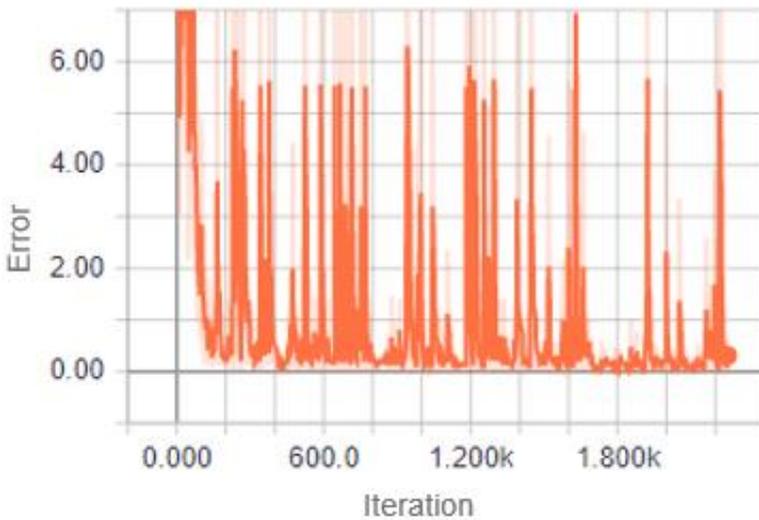


Gambar 5.18 Kesalahan RPN Regression sebesar 0.1014

- **Pengujian 150 Epoch model ResNet50.**



Gambar 5.19. Kesalahan RPN Classification sebesar 0.8116



Gambar 5.20. Kesalahan RPN Regression sebesar 0.6721

RIWAYAT PENULIS



Penulis yang mempunyai nama lengkap Ahmad Athfi Noor Laili merupakan mahasiswa Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya. Penulis merupakan anak sulung dari 3 bersaudara yang lahir di Demak pada tanggal 19 Februari 1996. Pada bulan juli 2015 penulis diterima di S1 Teknik Elektro melalui jalur Penerimaan Beasiswa Santri Berprestasi (PBSB) yang diselenggarakan oleh Kementerian Agama.

Selama kuliah penulis aktif di organisasi CSSMoRA ITS pada tahun 2017 sebagai kepala departemen Hubungan Luar (Hublu). Selain aktif di organisasi juga aktif sebagai asisten lab khususnya di Laboratorium Teknik Pengaturan AJ-104. Penulis berharap buku ini dapat dimanfaatkan sebagaimana mestinya. Dalam penulisan buku tugas akhir ini penulis mengharapkan kritik dan saran pembaca untuk perkembangan penelitian lebih lanjut bisa melalui email *athfielektroits@gmail.com*

