

PROPOSAL SKRIPSI
DETEKSI OBJEK DAN *INSTANCE SEGMENTATION* PARU-
PARU PADA CITRA X-RAY DADA MENGGUNAKAN MASK
R-CNN



Oleh:

M Afifudin Abdullah

190411100080

Dosen Pembimbing 1: Dr. Rima Tri Wahyuningrum, S.T, M.T.
Dosen Pembimbing 2: Dr. Indah Agustien Siradjuddin, S.Kom, M.Kom.

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
2022

LEMBAR PENGESAHAN

Nama : M Afifudin Abdullah

NIM : 190411100080

Bidang Minat : Kecerdasan Komputasional

Program Studi : Teknik Informatika

Dosen : 1. Dr. Rima Tri Wahyuningrum, S.T, M.T.
Pembimbing 2. Dr. Indah Agustien Siradjuddin, S.Kom, M.Kom.

Judul Tugas : Deteksi Objek dan *Instance Segmentation* Paru-Paru
Akhir Pada Citra *X-Ray* Dada Menggunakan *Mask R-CNN*

Proposal ini telah disetujui di seminar pada:

Tanggal, 2022

Dosen Pembimbing 1

Dosen Pembimbing 2

Dr. Rima Tri Wahyuningrum, S.T

M.T.

NIP. 19800820 200312 2 001

Dr. Indah Agustien Siradjuddin,

S.Kom, M.Kom

NIP. 19780820 200212 2 001

Mengetahui,

Koordinator Program Studi Informatika

Dr. Fika Hastarita Rachman, S.T, M.Eng

NIP. 19830305 200604 2 002

ABSTRAK

Virus corona adalah salah satu virus berbahaya di dunia. Virus ini bisa menyebabkan gangguan pernafasan seperti *Severe Acute Respiratory Syndrome* (SARS-CoV). Virus yang bisa menyebabkan seseorang terkena penyakit COVID-19 ini mempunyai variasi. Jika virus ini bisa menyerang paru-paru dan memicu peradangan, maka bisa menyebabkan penyakit Pneumonia. Untuk membedakan COVID-19 dan juga Pneumonia diperlukan evaluasi yang komprehensif seperti tes darah, tes swab PCR, sampai rontgen. Namun karena peralatan dan syarat yang dibutuhkan untuk beberapa tes yang disebutkan tadi terlalu banyak dan sulit, oleh karenanya diperlukan alternatif lain ataupun metode yang lebih cepat untuk pendeteksian COVID-19 ini. Salah satu caranya adalah dengan menggunakan *artificial intelligence* (AI).

Artificial intelligence (AI) adalah teknik yang dipakai pada citra *x-ray* paru-paru pada dada untuk deteksi penyakit yang lebih akurat dan proposal ini bertujuan untuk membangun proses diagnosa COVID-19 menggunakan teknik *deep learning* pada citra *x-ray* dada. Proposal ini menggunakan *Mask R-CNN* untuk melatih dan juga mengetes *dataset* untuk klasifikasi dan segmentasi pada citra *x-ray* paru-paru pada pasien pengidap penyakit COVID-19, Pneumonia dan pasien dengan paru-paru normal. Setelah sistem terbentuk, maka akan dilakukan evaluasi dari beberapa percobaan untuk mencari akurasi terbaik yang nantinya akan digunakan untuk pendeteksian COVID-19.

Kata kunci : COVID-19, Deep Neural Network, Deteksi Objek, *Mask R-CNN*, *Instance Segmentation*, R-CNN.

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.2.1. Permasalahan.....	3
1.2.2. Solusi Permasalahan.....	3
1.2.3. Pertanyaan Penelitian	3
1.3. Batasan Masalah	3
1.4. Manfaat dan Tujuan.....	4
1.4.1. Tujuan	4
1.4.2. Manfaat	4
1.5. Sistematika Proposal	4
BAB II KAJIAN PUSTAKA	5
2.1. Citra <i>X-Ray</i>	5
2.2. Deteksi Objek	6
2.3. Segmentasi Citra.....	7
2.4. <i>Mask Region-based Convolutional Neural Network (Mask R-CNN)</i>	8
2.4.1. <i>Convolutional Layer</i>	9
2.4.2. <i>Pooling Layer</i>	10
2.4.3. ReLU	10
2.4.4. Sigmoid	11
2.4.5. Softmax	11
2.4.6. <i>Region Proposal Network (RPN)</i>	12
2.4.7. <i>Non-Maximum Suppression (NMS)</i>	13
2.4.8. RoIAlign.....	13
2.4.9. <i>Backbone Architecture (ResNet-101)</i>	14
2.4.10. <i>Head Architecture</i>	15

2.4.11. Mask Branch (Fully Convolutional Network (FCN)).....	16
2.5. Loss Function	17
2.6. Evaluation Metrics	18
2.6.1. Intersection over Union (IoU).....	18
2.6.2. mean Average Precision (mAP).....	19
2.7. Penelitian Terkait	20
BAB III METODE USULAN.....	25
3.1. Dataset.....	25
3.2. Arsitektur Diagram <i>Training</i> dan <i>Testing</i>	26
3.3. Skenario Uji Coba	27
3.4. Perkiraan Jadwal.....	28
DAFTAR PUSTAKA	29

DAFTAR GAMBAR

Gambar 2. 1 Citra x-ray	5
Gambar 2. 2 Deteksi objek.....	6
Gambar 2. 3 Deteksi: <i>Object, semantic, instance</i>	7
Gambar 2. 4 Arsitektur <i>Mask R-CNN</i>	8
Gambar 2. 5 Konvolusi citra 3x3 dengan kernel 2x2.....	9
Gambar 2. 6 MaxPooling	10
Gambar 2. 7 Fungsi aktivasi ReLU	11
Gambar 2. 8 Fungsi aktivasi sigmoid.....	11
Gambar 2. 9 <i>Region Proposal Network (RPN)</i>	12
Gambar 2. 10 RoIAlign.....	13
Gambar 2. 11 <i>Building block residual learning</i>	15
Gambar 2. 12 Arsitektur ResNet-101	15
Gambar 2. 13 <i>Head Architecture</i>	16
Gambar 2. 14 FCN	16
Gambar 2. 15 Ilustrasi IoU.....	18
Gambar 3. 1 Contoh data <i>x-ray</i> beserta <i>ground-truth</i>	25

DAFTAR TABEL

Tabel 1 Penelitian terkait	20
Tabel 2 Pembagian data	25
Tabel 3 Tabel perkiraan jadwal.....	28

BAB I PENDAHULUAN

1.1. Latar Belakang

Pada Desember 2019, SARS-CoV-2 (nama sebelum Covid-19 (virus corona)) mengakibatkan pandemi di Wuhan, Provinsi Hubei, China [1]. Menurut *update* terakhir dari *World Health Organization* (WHO), secara global sampai pada tanggal 7 Desember 2022 sudah ada 642.379.243 kasus Covid-19 yang sudah dikonfirmasi, termasuk 6.642.118 diantaranya adalah kasus yang berujung kematian [2]. Covid-19 dapat menyebabkan beberapa infeksi seperti gangguan pernafasan, demam, batuk, sesak napas, dan kesulitan bernapas. Lebih parahnya, Virus ini bisa menyebabkan penyakit pneumonia, sindrom pernapasan akut, gagal ginjal, hingga kematian [3].

Beberapa metode diagnosis sudah pernah digunakan untuk mendeteksi Covid-19 secara klinis. Metode deteksi yang sering digunakan untuk mendeteksi Covid-19 secara klinis adalah *real-time reverse transcriptase polymerase chain reaction* (RT-PCR) [4]. Namun meskipun RT-PCR adalah teknik standar yang digunakan untuk mendeteksi Covid-19, metode ini juga mempunyai keterbatasan yang menghasilkan deteksi yang tidak tepat. Salah satunya adalah sensitifitasnya yang rendah dan juga waktu yang dibutuhkan untuk menganalisis tanpa automasi sistem bisa mencapai 24 jam untuk mendapatkan hasilnya. Hasil ini sangat dipengaruhi oleh pra-analisis dan penanganan sampel yang membutuhkan perhatian khusus [5]. Dikarenakan beberapa keterbatasan tersebut, diperlukan *artificial intelligence* (AI) sebagai sistem kesehatan yang klinis yang bisa mendeteksi Covid-19 secara cepat dan akurat.

Teknik *deep learning* sendiri sudah bangkit ditahun-tahun sebelumnya dan ikut berkontribusi dalam banyak skenario riset. Khususnya di bagian citra medis, data citra medis seperti *X-ray* dada bisa menyediakan akurasi yang tinggi menggunakan *deep learning*. Seperti yang diketahui, mesin *X-ray* menghasilkan hasil pemindaian yang murah dan cepat untuk memindai berbagai organ manusia. Interpretasinya sendiri akan dilakukan secara manual oleh para ahli. Data *X-ray* dada bisa dilatih oleh *deep learning* agar membantu para ahli medis dalam mendeteksi pasien Covid-19 [6]–[8].

Tujuan penelitian ini adalah menghasilkan model yang memiliki kinerja ataupun akurasi terbaik dalam melakukan deteksi otomatis Covid-19 menggunakan algoritma *Mask R-CNN* dengan menggunakan pemilihan *Intersection over Union* (IoU) yang tepat. Model pada penelitian ini menggunakan masukan berupa data *X-ray* dada digital dengan skenario pengaturan *threshold* IoU untuk menghasilkan model dengan akurasi yang beragam.

Pada tahun 2020, Wang, *et al.* melakukan riset untuk membangun sebuah model yang bernama COVID-Net dengan masukan data berupa data *X-ray* dada digital. Penelitian tersebut berhasil memperoleh akurasi sebesar 92.4% dikeseluruhan kelas [9]. Kemudian pada tahun 2021, Gaur, *et al.* melakukan penelitian untuk mendapatkan model dengan terbaik untuk deteksi Covid-19 dan ditemukan bahwa algoritma VGG16 berhasil mendapatkan akurasi sebesar 87.84%, kemudian InceptionV3 memperoleh akurasi sebesar 91.32% disusul oleh EfficientNetB0 yang memperoleh akurasi sebesar 92.93% [10]. Kemudian pada tahun yang sama, Podder, *et al.* melakukan riset menggunakan algoritma *Mask R-CNN* untuk klasifikasi pasien dan non-pasien Covid-19. Penelitian tersebut memperoleh akurasi sebesar 96.98% [11]. Namun pada penelitian tersebut, eksperimen yang digunakan hanyalah berfokus pada *backbone* CNN yang digunakan (dimana menggunakan ResNet41, ResNet50, ResNet65 dan ResNet101). Namun *threshold* yang digunakan untuk menghasilkan *bounding box* deteksi tidak disebutkan dalam eksperimen dimana dalam paper sendiri menggunakan nilai tetap dari algoritma *Mask R-CNN* yaitu 0.5 [12].

Sehingga, dengan harapan agar mendapatkan hasil yang lebih baik, pada penelitian ini akan dilakukan uji coba ataupun eksperimen terhadap *Intersection over Union* (IoU), dimana IoU memegang peranan penting pada hasil *bounding box* deteksi objek. Dimana ketika ada sebuah citra, IoU menghitung tingkat kemiripan area prediksi dengan area yang sebenarnya (*ground truth*), yang didefinisikan sebagai irisan dari kedua area dibagi dengan gabungan (*union*) keduanya. Dimana *bounding box* hasil seleksi IoU ini yang akan mempengaruhi proses deteksi objek nanti.

Berdasarkan hal tersebut, proposal ini bertujuan membangun sebuah model *deep learning* untuk melakukan deteksi objek terhadap citra *X-Ray* dada. *Backbone*

architecture yang dipakai adalah ResNet-101 dengan *region proposal* yang diperoleh dari *Region Proposal Network*. Selain itu *threshold* IoU yang dipakai dalam uji coba ini juga beragam dengan nilai 0.4, 0.5, 0.6, dan 0.7. Tujuan dari penelitian ini adalah untuk mengetahui seberapa besar pengaruh IoU *threshold* terhadap tingkat akurasi yang dihasilkan model.

1.2. Rumusan Masalah

1.2.1. Permasalahan

Bagaimana implementasi *deep learning* untuk dapat menghasilkan sistem yang akurat pada deteksi Covid-19 serta bagaimana pengaruh IoU *threshold* yang digunakan dalam proses deteksi Covid-19?

1.2.2. Solusi Permasalahan

Solusi yang diusulkan untuk mendapatkan sistem dengan kinerja terbaik adalah *Region-based Convolutional Neural Network* dengan menggunakan *Mask R-CNN* untuk menghasilkan akurasi yang tinggi. Serta akan dilakukan skenario uji coba terhadap IoU *threshold* yang digunakan untuk mengetahui seberapa pengaruh nilai IoU terhadap hasil deteksi yang dihasilkan.

1.2.3. Pertanyaan Peneliti

Pertanyaan peneliti digunakan untuk mengetahui apa yang akan dilakukan pada penelitian ini sehingga memperoleh hasil yang sesuai yaitu:

Bagaimana pengaruh IoU *threshold* yang digunakan terhadap tingkat akurasi yang dihasilkan dalam proses deteksi menggunakan algoritma *Mask R-CNN*?

1.3. Batasan Masalah

1. *Dataset* difokuskan pada citra *x-ray* dada dari radiografi,
2. *Dataset* dalam ujicoba ini terdiri dari tiga kelas, yaitu Normal, Pneumonia dan Covid-19,
3. *Dataset* citra *x-ray* dada yang digunakan dalam penelitian ini adalah dataset publik bersumber dari Qatar University yang tersedia dalam repositori Kaggle dengan situs: <https://www.kaggle.com/datasets/anasmohammedtahir/covidqu>,
4. *Dataset* berjumlah 1500 dengan 500 diantaranya Covid, 500 Pneumonia dan 500 lainnya normal.
5. *Dataset* berukuran 256x256 piksel.

6. *Dataset* dibagi 90% *training* dan 10% *testing*
7. *Dataset* dilakukan K-Fold dengan jumlah k adalah 5.
8. Model menggunakan *Mask R-CNN* dengan *learning rate* 0.02, *optimizer* AdamW, *batch size* 16 dan *epoch* 100

1.4. Manfaat dan Tujuan

1.4.1. Tujuan

Tujuan penelitian ini digunakan untuk menjawab pertanyaan peneliti yaitu:

1. Mengetahui pengaruh nilai *IoU threshold* terhadap tingkat akurasi yang dihasilkan dalam proses deteksi menggunakan algoritma *Mask R-CNN*.

1.4.2. Manfaat

Hasil dari penelitian ini dapat dijadikan rujukan untuk membangun model deteksi untuk menentukan status pasien Covid-19 secara cepat dan akurat dari citra *x-ray* dada yang lebih baik.

1.5. Sistematika Proposal

Sistematika penulisan proposal tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang mengenai ruang lingkup masalah yang dibahas, solusi yang diusulkan untuk menyelesaikan permasalahan, tujuan serta manfaat dengan diterapkannya solusi yang diusulkan, dan sistematika penulisan proposal.

BAB II KAJIAN PUSTAKA

Bab ini berisikan pembahasan teori terkait *Deep Learning Convolutional Neural Network*, *Mask R-CNN* sebagai deteksi dan *library* yang akan digunakan, serta penelitian terkait yang telah ada sebelumnya.

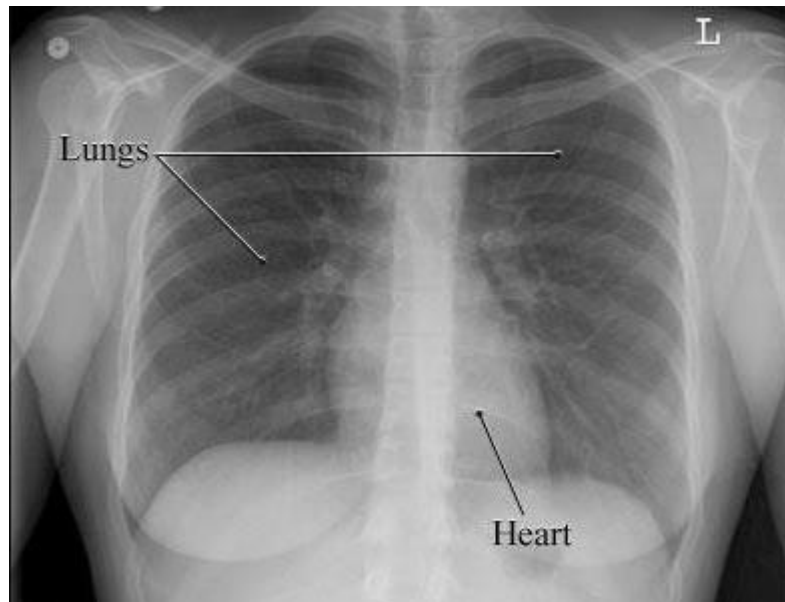
BAB III METODE USULAN

Pada bab ini menjelaskan tentang *dataset* yang akan digunakan, *pre-processing*, arsitektur sistem, arsitektur *training* dan *testing*, skenario uji coba, dan jadwal pengerjaan.

BAB II KAJIAN PUSTAKA

2.1. Citra *X-Ray*

X-Ray (Sinar-X/Rontgen) adalah jenis radiasi yang disebut dengan gelombang elektromagnetik. *x-ray* membuat citra dari sebuah bagian dalam tubuh. Citra ini menampilkan bagian dari sebuah tubuh dengan nuansa hitam dan juga putih. Ini karena jaringan yang berbeda juga menyerap jumlah radiasi yang berbeda juga. Kalsium dalam tulang paling banyak menyerap *x-ray*, oleh karenanya hasil citra tulangnya terlihat putih. Lemak dan juga jaringan-jaringan lembut/tipis lainnya akan tampil keabuan. Udara tidak terlalu menyerap radiasi *x-ray*, oleh karenanya hasil citra paru-paru terlihat hitam [13], [14].



Gambar 2. 1 Citra x-ray

Jika sebuah *x-ray* menembus badan dan sampai ke *x-ray detector*, maka citra akan terbentuk yang merepresentasikan bayangan yang terbuat dari objek yang ada didalam tubuh. Salah satu tipe dari *x-ray detector* adalah film fotografi, namun masih ada banyak juga tipe dari *detector* yang bisa menghasilkan citra digital. Citra *x-ray* yang terbentuk dengan proses ini disebut dengan radiografi [15].

Ketika seseorang akan dikenakan *x-ray*, maka seseorang tersebut harus menggunakan proteksi seperti celemek timah untuk menutupi dan melindungi bagian-bagian tertentu dari tubuh. Jumlah radiasi yang diterima dari *x-ray* sangatlah kecil. Seperti contoh, *x-ray* dada hanya memberikan dosis radiasi yang sama pada saat tubuh berada di lingkungan selama 10 hari. Penggunaan paling umum dari

sebuah *x-ray* adalah untuk mengecek patah tulang, tapi *x-ray* juga bisa diterapkan pada hal lain. Seperti contoh, *x-ray* dada dapat memeriksa pneumonia. Mammograms menggunakan *x-ray* untuk mencari kanker payudara [13].

2.2. Deteksi Objek

Deteksi objek adalah cara untuk secara presisi mengestimasi konsep dan lokasi dari sebuah objek yang ada di setiap citra untuk mendapatkan pemahaman lebih mengenai citra. Cabang dari deteksi objek juga meliputi deteksi wajah, deteksi pejalan kaki, dan deteksi tengkorak. Masalah yang ada pada deteksi objek adalah pada saat menentukan dimana lokasi objek (*object localization*) berdasarkan citra yang sudah diberikan dan termasuk di kategori apakah objek tersebut berada (*object classification*). Oleh karenanya *pipeline* dari deteksi objek tradisional bisa dipecah menjadi tiga bagian: seleksi daerah (*region*) informatif, ekstraksi fitur (*feature extraction*) dan klasifikasi.

Beberapa objek yang berbeda bisa tampil di segala posisi dari sebuah citra dan juga bisa memiliki aspek rasio dan ukuran yang berbeda. Oleh karenanya, wajar jika memilih untuk memindai semua bagian dari citra dengan *sliding window* yang *multi-scale*. Dikarenakan banyaknya kandidat *window*, secara komputasi sangatlah berat dan memproduksi banyak *window* yang redundan. Meskipun demikian, jika jumlah *sliding window* sudah ditetapkan, maka daerah yang akan diproduksi juga bisa tidak memuaskan.



Gambar 2. 2 Deteksi objek

Untuk mengenali objek yang berbeda, diperlukannya ekstraksi fitur visual yang bisa merepresentasikan citra secara baik. Disamping itu, *classifier* (pengklasifikasi) diperlukan untuk membedakan objek target dengan semua

kategori yang lain untuk membuat representasi lebih semantik dan informatif untuk pengenalan visual [16], [17].

2.3. Segmentasi Citra

Segmentasi citra, yang biasanya didefinisikan sebagai pemisahan sebuah citra menjadi beberapa wilayah (*region*) adalah salah satu langkah untuk analisis citra. Tujuan dari segmentasi citra adalah untuk memisahkan citra menjadi beberapa wilayah (*region*) yang sangat bisa digunakan atau diproses untuk pekerjaan (*task*) tertentu. Salah satunya bisa berupa deteksi dari sebuah organ seperti jantung, hati, paru-paru yang citranya didapatkan dari pemindaian (*scan*) CT ataupun MR.

Pendekatan Segmentasi citra bisa dikategorikan berdasarkan fitur dan teknik yang digunakan. Fitur bisa berupa intensitas piksel, besaran gradien, ukuran ataupun tekstur. Sedangkan teknik segmentasi yang dipakai bisa dikategorikan menjadi tiga yaitu *region-based* (berbasis wilayah), *edge-based* (berbasis tepian), dan *classification* (klasifikasi). *Region-based* dan *edge-based* melakukan segmentasi berdasarkan kemiripan wilayah dan perbedaan wilayah dari beberapa fitur, sedangkan teknik klasifikasi memberi label kelas untuk setiap pikselnya atau voxel berdasarkan nilai fitur [18].

Salah satu dari pendekatan simpel untuk segmentasi citra adalah dengan pendekatan level intensitas atau yang biasa disebut dengan pendekatan *threshold*. Pendekatan ini memisahkan sebuah citra menjadi dua kelas dan memasukkan piksel dengan nilai tertentu ke dalam sebuah kelas dan sisa piksel lainnya dimasukkan ke kelas lain. Piksel yang melewati batas *threshold* akan dianggap sebuah piksel objek dan piksel tersebut akan dimasukkan nilai 1 dan piksel lainnya akan dianggap sebuah latar belakang dan akan dimasukkan nilai 0.



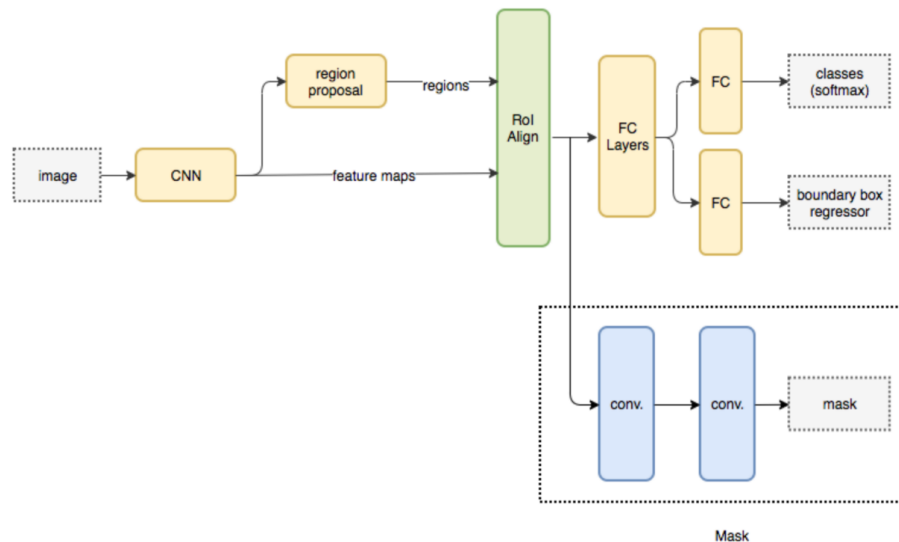
Gambar 2. 3 Deteksi: *Object, semantic, instance*

Discontinuity adalah salah satu pendekatan yang memanfaatkan variasi intensitas yang ada di sebuah piksel. Jika sebuah citra terdapat dua atau lebih

batasan (*boundaries*) objek maka bisa dilakukan segmentasi. Batas-batas objek yang dimaksud adalah pembentukan tepi dimana tepi disini adalah perubahan intensitas suatu piksel dengan piksel terdekatnya secara mendadak [19].

Semantic segmentation dan *object detection* sudah dipelajari dengan bagus untuk menyelesaikan masalah lokalisasi objek dikarenakan berkembangnya *deep learning* di dalamnya. Namun, *semantic segmentation* yang akan melabeli setiap piksel citra yang ada dengan kelas dari objeknya tidak bisa membedakan objek yang berbeda dengan kelas yang sama seperti pada contoh gambar 2.3. *Object detection* melokalisasi objek yang berbeda namun masih melakukannya dengan cara membuat kotak yang membungkus objek. *Instance segmentation* melokalisasi objek pada tingkatan piksel seperti pada contoh gambar 2.3. Dan bisa juga memisahkan dua objek yang saling tumpang tindih. Tidak seperti pendahulunya (*semantic segmentation*), *instance segmentation* tahu bagaimana objek yang berbeda pada kelas yang sama dan beroperasi pada tingkat piksel [20].

2.4. Mask Region-based Convolutional Neural Network (Mask R-CNN)



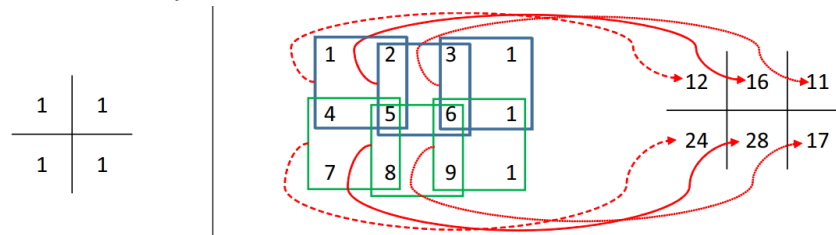
Gambar 2. 4 Arsitektur *Mask R-CNN*

Mask R-CNN adalah pengembangan dari *Faster R-CNN* yang menambahkan cabang untuk prediksi *segmentation mask* di setiap *Region of Interest* (RoI), secara paralel dengan cabang yang sudah ada untuk klasifikasi dan juga *bounding box regression*. Cabang untuk prediksi *mask* adalah FCN kecil yang diaplikasikan di setiap RoI yang akan memprediksi *mask* secara piksel ke piksel.

Faster R-CNN mempunyai dua kandidat *output* yaitu *class label* dan juga *bounding box*. yang kemudian akan kita tambahkan cabang yang ketiga yaitu untuk *object mask* [21].

Pada dasarnya *Mask R-CNN* mempunyai tiga tahapan, yang pertama adalah *backbone network* yang akan mengekstraksi *feature maps* dari citra input. Kedua, *feature maps* yang sudah didapatkan tadi dari *backbone network* akan dimasukkan ke dalam *Region Proposal Network* (RPN) untuk menghasilkan *Region of Interest* (RoI) dengan menggunakan *Fully Convolutional Network* (FCN). Ketiga, *Region of Interest* (RoI) tadi yang dihasilkan dari FCN akan dipetakan untuk ekstraksi fitur target yang sesuai dengan *share feature maps*. Kemudian akan diproses oleh *Fully Connected Layer* (FC) untuk menghasilkan klasifikasi target dan *Fully Convolutional Network* (FCN) untuk menghasilkan *instance segmentation*. Semua proses ini digunakan untuk memperoleh tiga *output* yaitu nilai klasifikasi, *bounding box* dan *segmentation mask* [22].

2.4.1. Convolutional Layer

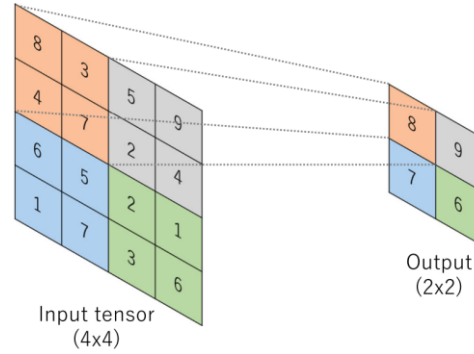


Gambar 2. 5 Konvolusi citra 3x3 dengan *kernel* 2x2

Dimulai dari konvolusi matriks dengan satu kernel. Anggap citra input sebesar 3x4 dan ukuran kernel adalah 2x2. Jika citra input ditimpa dengan kernel, maka komputasi antara nilai pada lokasi yang sama bisa dilakukan dan akhirnya akan mendapatkan satu nilai dengan cara menjumlahkan semua nilai pada setiap posisi. Sebagai contoh jika kernel menimpa citra input pada lokasi atas samping kiri, hasil konvolusi spasial di lokasi tersebut adalah: $1 \times 1 + 1 \times 4 + 1 \times 2 + 1 \times 5 = 12$. Kemudian kernel perlu digeser kebawah sebanyak satu piksel dan hasil konvolusi pada lokasi tersebut adalah: $1 \times 4 + 1 \times 7 + 1 \times 5 + 1 \times 8 = 24$. Kernel perlu digeser lagi kebawah sampai pada batas bawah pada citra input. Setelahnya, kernel perlu dikembalikan pada posisi awal (atas) namun kali ini digeser kekanan sebanyak satu

piksel. Pergeseran ini dilakukan sampai semua lokasi dari citra input sudah ditimpa kernel, seperti pada gambar 2.4.1 [23].

2.4.2. Pooling Layer



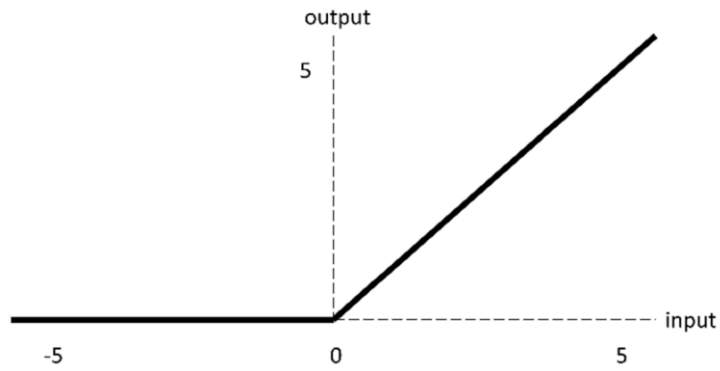
Gambar 2. 6 MaxPooling

Pooling layer bermanfaat untuk mereduksi dimensi, dan disaat bersamaan berarti mereduksi jumlah parameter dan kompleksitas dari sebuah model [24]. Tidak ada parameter yang bisa dilatih pada *pooling layer*, dimana ukuran filter, *stride*, dan *padding* adalah *hyperparameter* pada proses *pooling*. Bentuk *pooling* paling populer adalah *max pooling*, yang akan mengekstraksi *feature map input* dan akan menghasilkan *output* berupa nilai maksimum pada setiap area yang ditimpa filter *pooling layer* ini, kemudian mengabaikan nilai yang lain. *Max pooling* dengan ukuran filter 2x2 dengan *stride* 2 biasanya paling sering digunakan. Gambar 2.6 adalah Contoh dari *Max Pooling* dengan ukuran filter 2x2, tanpa padding, *stride* 2, yang akan mengekstraksi 2x2 *patches* dari input, dan menghasilkan *output* berupa nilai maksimum masing-masing *patch*, dan akan mengabaikan nilai yang lain, menghasilkan *downsampling* pada dimensi dari input dengan faktor 2 [25].

2.4.3. ReLU

Tujuan utama pada proses *nonlinearity* adalah untuk menyesuaikan atau memotong hasil *output*. Beberapa fungsi non-linear bisa dipakai untuk CNN. Namun, ReLU (*Rectified Linear Unit*) adalah salah satu fungsi non-linear yang sering dipakai. Fungsi aktivasi ReLU bisa didefinisikan sebagai [26][27]:

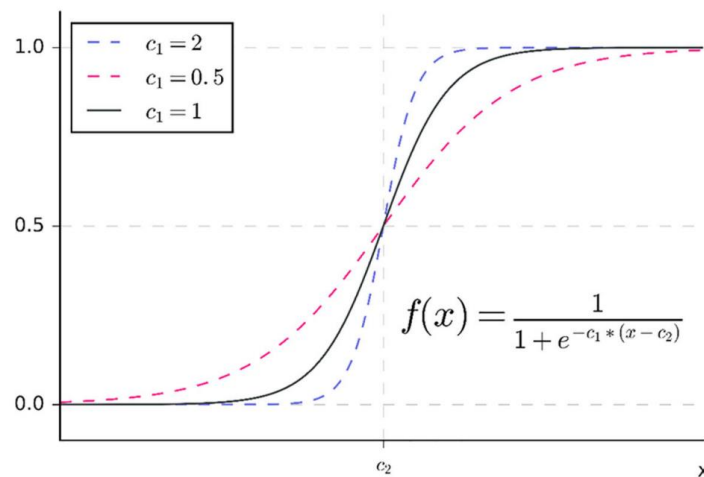
$$ReLU = \max(0, x) \quad (1)$$



Gambar 2. 7 Fungsi aktivasi ReLU

2.4.4. Sigmoid

Fungsi sigmoid adalah sebuah fungsi yang membentuk seperti huruf S yang dimulai dengan kemiringan nol atau mendekati nol kemudian berlanjut dengan kemiringan positif dan kembali lagi ke nol [28].



Gambar 2. 8 Fungsi aktivasi sigmoid

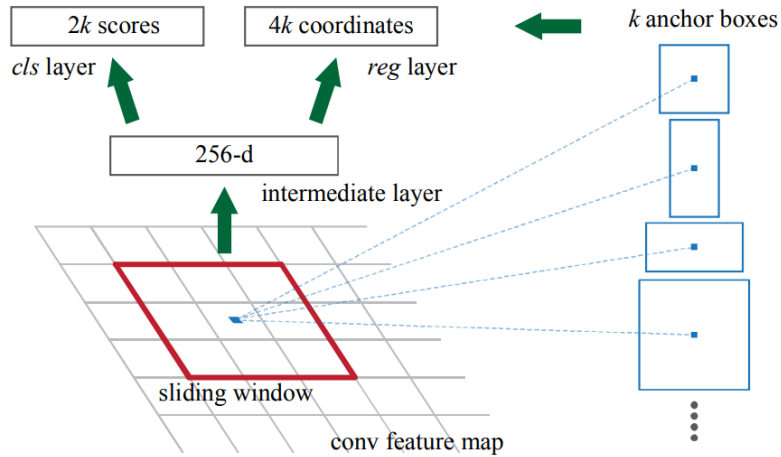
2.4.5. Softmax

Fungsi ini membedakan dan merepresentasikan probabilitas dari sebuah *output*. Fungsi ini mengaplikasikan perubahan normalisasi eksponensial dari nilai *output* dengan persamaan:

$$o_i = \frac{e^{z_i}}{\sum_{i=1}^M e^{z_i}} \quad (2)$$

dimana o_i adalah hasil *output*, i , z_i adalah *output* i sebelum *softmax*, dan M adalah jumlah *output* [26].

2.4.6. Region Proposal Network (RPN)



Gambar 2. 9 Region Proposal Network (RPN)

Region Proposal Network (RPN) mengambil input dari citra dengan berbagai ukuran dan akan menghasilkan *output* berupa kotak proposal objek dan juga nilai skor keobjektivitasannya. Model yang dipakai untuk memproses ini adalah FCN (*Fully Convolutional Network*). Untuk menghasilkan *region proposal*, *network* kecil harus di-*slide* di atas hasil dari *convolutional feature map* yang merupakan *output* dari layer konvolusi terakhir dari modul CNN yang di-sharing. Network ini mengambil input dengan ukuran $n \times n$ *spatial window* dari *feature map*. Setiap *window*-nya akan dipetakan menjadi fitur dengan dimensi rendah dan akan diproses di *box-regression layer* dan juga *box-classification layer*. Banyak *region proposal* yang diprediksi dari setiap lokasi *sliding-window*. Jumlah maksimum dari banyaknya proposal yang bisa diprediksi dari setiap lokasi *sliding-window* adalah k . Layer *reg* berfungsi untuk menghitung skor/nilai keobjektivitasan. Setiap jumlah k proposal di parameterisasi secara relatif kepada box referensi k yang disebut *anchor*.

Sebuah *anchor* ada di tengah dari *sliding window*. Modul ini menggunakan *translation-invariant* untuk mereduksi fitur dari ukuran. Modul ini juga menggunakan *multi-scale anchor* sebagai referensi objek regresi untuk menggunakan berbagai skala. *bounding-box* untuk klasifikasi dan regresi ini mempunyai referensi pada *anchor boxes* dengan *multiple scale* dan rasio aspek. Secara bawaan, 3 skala dengan 3 rasio aspek digunakan, yang menjadikan $k = 9$

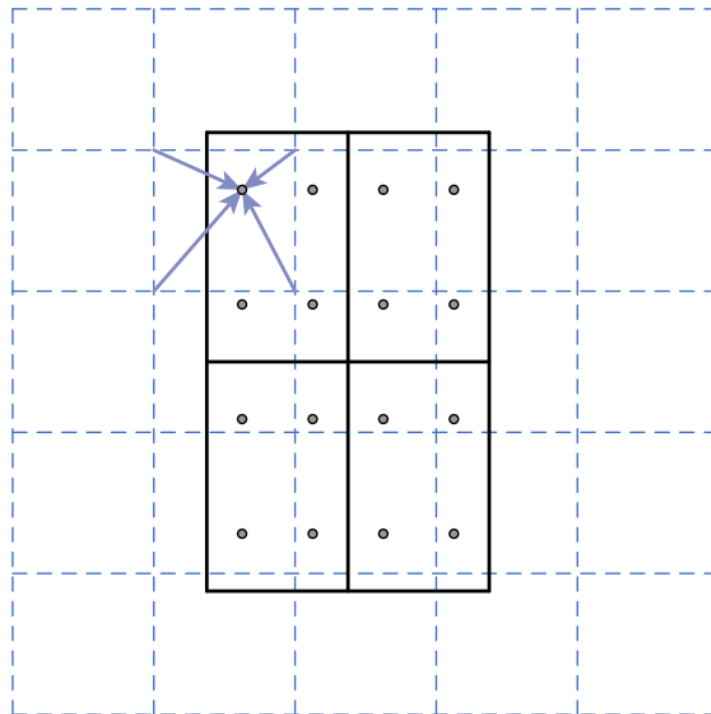
anchor setiap *sliding window*. Untuk *anchor*, digunakan *box area* dari 128^2 , 256^2 , dan 512^2 piksel dengan 3 aspek rasio dari 1:1, 1:2, dan 2:1 [29].

2.4.7. *Non-Maximum Suppression* (NMS)

Semua *modern object detectors* mengikuti tiga tahapan berikut: (1) mengajukan proposal dari setiap *windows* (dengan menggunakan *sliding windows* biasanya), (2) menilai (skor) *window* dengan *classifier/regressor*, dan (3) menggabungkan semua *window* yang mendeteksi objek yang sama. Pada tahapan terakhir, proses ini biasanya dinamakan dengan “*Non-Maximum Suppression*” [30].

Standar dari NMS sendiri terdiri dari secara “rakus” menggabungkan *window* dengan skor deteksi terbaik dengan *window* dengan skor deteksi lebih rendah jika *window* dengan skor deteksi lebih rendah *overlap* (seperti $\text{IoU} > 0.5$), yang akan disebut dengan “*GreedyNMS*” [31].

2.4.8. RoIAlign



Gambar 2. 10 RoIAlign

RoIPool [29] adalah operasi standar dari ekstraksi fitur dari setiap RoI [12]. RoIPool menggunakan MaxPooling untuk mengkonversi fitur dari RoI menjadi fitur yang lebih kecil lagi dengan batas spasial tetap dari $H \times W$ (contoh 7×7), dimana H dan W adalah *hyperparameter* dari layer yang independen dari RoI. RoI

adalah *rectangular window* yang ada di dalam *convolutional feature map*. Setiap RoI didefinisikan dari r, c, h, w dimana r dan c adalah titik atas kiri pojok dan h dan w adalah *height* (tinggi) dan *width* (lebar). RoIPool pertama-tama akan mengkuantisasi *floating-number* dari sebuah RoI menjadi ukuran yang diskrit (tanpa koma). RoI ini kemudian akan dibagi lagi menjadi beberapa bagian yang setiap dari *sub-RoI* ini akan dikuantisasi juga dan terakhir nilai fitur dari setiap *sub-RoI* akan diagregasi menjadi satu (biasanya memakai *max-pooling*). Kuantisasi dilakukan pada sebuah koordinat seperti contoh x dengan menghitung $[x/16]$, dimana 16 adalah *stride* dari *feature map* dan $[.]$ adalah *rounding* (pembulatan). Kuantisasi ini menghasilkan *misalignment* dari RoI dengan fitur yang sudah diekstraksi. Gambar 2.10 adalah contoh RoIAlign, garis putus-putus adalah gambaran dari *feature map* dan garis dengan warna solid adalah RoI (contoh 2x2). Setiap titik akan diinterpolasi di masing-masing kotak.

RoIAlign merubahnya dengan cara meniadakan kuantisasi dari RoI (seperti contoh menggunakan $x/16$ dibandingkan $[x/16]$). Kemudian menggunakan *bilinear interpolation* untuk mengkalkulasi nilai pasti dari setiap fitur input dan akan diagregasi (*max* atau *average*) [32]. Rumus dari *bilinear interpolation* adalah:

$$P = \frac{y_2 - y}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} Q_{11} + \frac{x - x_1}{x_2 - x_1} Q_{21} \right) + \frac{y - y_1}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} Q_{12} + \frac{x - x_1}{x_2 - x_1} Q_{22} \right) \quad (3)$$

dimana Q_{11}, Q_{21}, Q_{12} , dan Q_{22} adalah titik data (pangkal simbol panah dari gambar 2.4.8) dan P adalah titik yang akan diinterpolasi [32].

2.4.9. Backbone Architecture (ResNet-101)

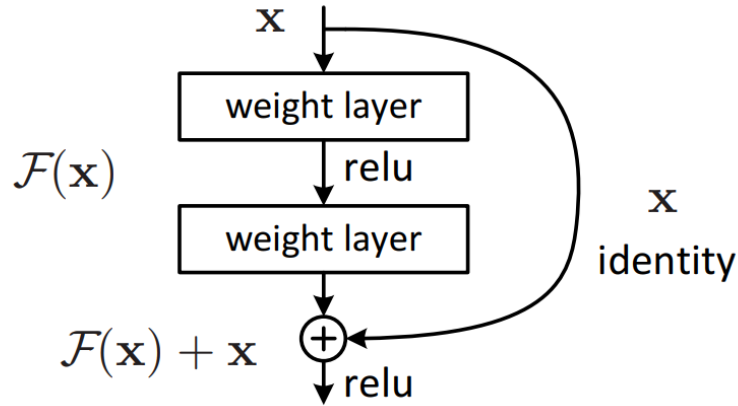
Residual learning menggunakan teknik yang disebut “*shortcut connections*”. *Shortcut connections* adalah teknik melewati satu atau lebih layer dengan melakukan *identity mapping* yang *outputnya* akan dimasukkan kedalam *stacked layer*. Dibandingkan dengan membiarkan setiap beberapa dari *stacked layer* untuk langsung dipetakan (*mapping*) dan berharap akan *fit*, secara eksplisit layer tersebut dimasukkan kedalam *residual mapping*. $H(x)$ adalah *initial mapping*, *stacked non-linear mapping* akan di-fit pada *initial mapping* menjadi $F(x) := H(x) - x$. *Mapping* original menjadi $F(x) + x$.

Building block didefinisikan oleh:

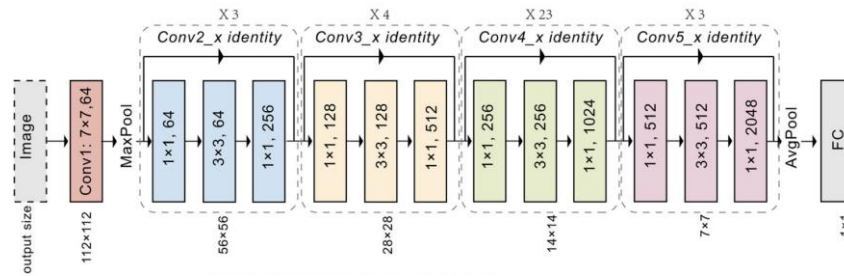
$$y = F(x, \{W_i\}) + x \quad (4)$$

x dan y adalah input dan *output* vektor dari sebuah layer. Fungsi $F(x, \{W_i\})$ adalah *residual learning* yang akan dipakai. Dimensi dari x dan F haruslah sama pada persamaan 4. Jika tidak, maka diperlukan proyeksi linear W_s dari dimensi *shortcut connection* untuk menyamakan dimensi [33], [34]:

$$y = F(x, \{W_i\}) + W_s x \quad (5)$$



Gambar 2. 11 Building block residual learning

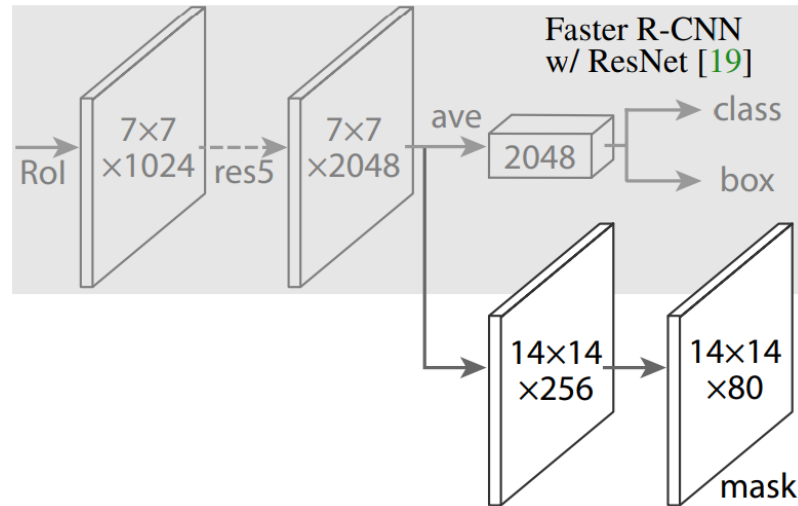


Gambar 2. 12 Arsitektur ResNet-101

Hasil dari *feature extraction* pada *stage* ke-4 dari konvolusi terakhir akan dipakai pada arsitektur ini.

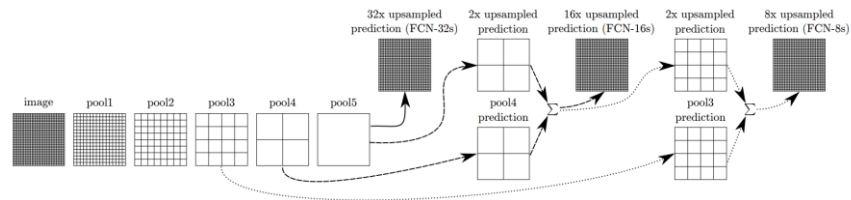
2.4.10. Head Architecture

Untuk *head architecture* disini memakai arsitektur bawaan dari *Faster R-CNN* dengan menyisipkan *stage* ke-5 dari ResNet [12]. Gambar 2.13 menampilkan *head architecture* dimana ini adalah ekstensi dari *Faster R-CNN* menggunakan ResNet-C4 (ekstraksi fitur terdapat pada layer konvolusi keempat). Ini disisipkan tepat sebelum *mask branch* ditambahkan. Semua konvolusinya adalah 3x3, kecuali *output* (1x1), dekonvolusi 2x2 dengan *stride* 2, dan memakai ReLU di *hidden layer*.



Gambar 2. 13 Head Architecture

2.4.11. Mask Branch (Fully Convolutional Network (FCN))



Gambar 2. 14 FCN

FCN dibuat untuk segmentasi dengan cara mengkombinasikan layer dari hirarki fitur dan memperbaiki presisi spasial dari *output*, lihat pada gambar 2.4.11. *Stride* 32 yang dilakukan pada layer terakhir membatasi detail dari hasil *upsampling output*. Hal ini bisa diatasi dengan membuat koneksi yang akan mengkombinasikan layer prediksi terakhir dengan layer awal dengan *stride* rendah (lihat pada gambar 2.4.11). Semakin sedikit piksel yang bisa dilihat maka prediksinya harusnya membutuhkan layer yang sedikit juga, jadi masuk akal jika membuatnya dari layer *output* yang lebih *shallow*. Pada gambar 2.14 terdapat Garis utuh (FCN-32s) yang akan *upsampling* prediksi dengan *stride* 32 menjadi piksel seperti input. Garis putus-putus (FCN-16s) mengkombinasikan prediksi dari layer terakhir dengan layer pool4 dengan *stride* 16. Garis titik-titik (FCN-8s) akan menjadi tambahan prediksi dari pool3 dengan *stride* 8.

Pertama bagi *output* dari *stride* menjadi dua dengan memprediksi dari layer *stride* 16 piksel. Tambahkan konvolusi 1x1 diatas pool4 untuk memproduksi kelas prediksi tambahan. Gabungkan hasil prediksinya dengan prediksi conv7 dengan

stride 32 dengan menambahkan 2x *upsampling* dan *summing* di kedua prediksi. Inisialisasi 2x *upsampling* menjadi *bilinear interpolation*. Terakhir, prediksi dari *stride* 16 kemudian di *upsampling* kembali menjadi citra. Ini disebut FCN-16s. FCN-16s dilatih secara *end-to-end*, diinisialisasi dengan parameter yang terakhir, *coarser net*, yang akan menjadi FCN-32s. Parameter baru pada pool4 diinisialisasi dari nol, jadi *network* berjalan tanpa adanya modifikasi prediksi. Selanjutnya menggabungkan antara pool3 dengan 2x *upsampling* dari prediksi yang merupakan hasil gabungan dari pool4 dan conv7 akan membentuk FCN-8s [35].

2.5. Loss Function

Pada saat *training*, *multi-task loss* dari setiap RoI didefinisikan sebagai $L = L_{cls} + L_{box} + L_{mask}$. *Classification loss* L_{cls} dan *bounding-box loss* L_{box} identik satu dengan lainnya. *Mask branch* mempunyai K_m^2 dimensi *output* dari setiap RoI dimana K adalah *binary mask* dengan resolusi $m \times m$. Operasi sigmoid akan dioperasikan di setiap pikselnya dan menghitung L_{mask} dengan *average binary cross-entropy loss*. Setiap RoI yang terhubung dengan *ground-truth* dari sebuah kelas k , L_{mask} terdapat pada *mask* k (*output mask* yang lain tidak ikut dalam perhitungan *loss*) [12].

Untuk *training* RPN, akan dilakukan klasifikasi objek atau bukan objek pada setiap *anchor*-nya. Setiap *anchor* akan dinyatakan positif jika memiliki IoU yang lebih tinggi pada *ground-truth box*, atau *anchor* dengan IoU yang lebih tinggi nilainya dari 0.7 dengan *ground-truth box*. *Anchor* akan dinyatakan negatif jika memiliki IoU kurang dari 0.3. Dan sisa dari *anchor* yang bukan negatif ataupun positif akan diabaikan. *Loss function* didefinisikan sebagai berikut:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (6)$$

dimana i adalah *index* dari *anchor* dari setiap *mini-batch* dan p_i probabilitas dari prediksi keobjektivitasan *anchor* i . *Ground-truth* p_i^* bernilai 1 jika *anchor* adalah positif, dan 0 jika negatif. t_i adalah vektor yang merepresentasikan 4 koordinat dari prediksi *bounding-box*, dan t_i^* adalah *ground-truth box* dari positif *anchor*. *Classification loss* L_{cls} adalah *loss* dari dua kelas (objek dan non-objek) menggunakan *softmax loss* (*cross-entropy*). Untuk L_{reg} adalah $L_{reg} = R(t_i, t_i^*)$ dimana R adalah *smooth L_1 loss*[29]. *smooth L_1 loss* didefinisikan dari [36]:

$$\text{smooth } L_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{selainnya,} \end{cases} \quad (7)$$

$p_i^* L_{reg}$ berarti *regression loss* akan diaktivasi oleh *anchor* positif ($p_i^*=1$) dan tidak diaktivasi jika sebaliknya ($p_i^*=0$).

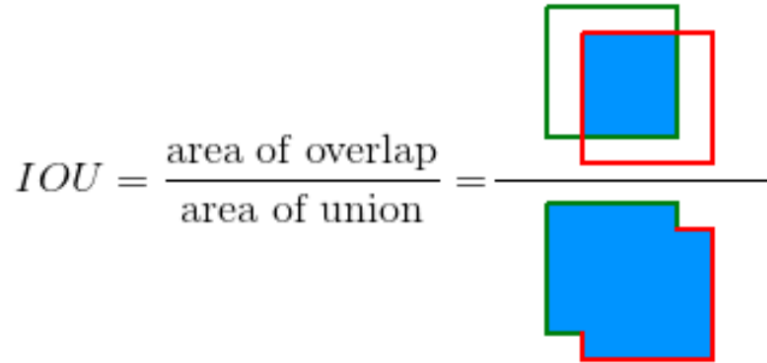
Untuk *bounding-box regression*:

$$\begin{aligned} t_x &= \frac{x - x_a}{w_a}, \quad t_y = \frac{y - y_a}{h_a}, \\ t_w &= \log\left(\frac{w}{w_a}\right), \quad t_h = \log\left(\frac{h}{h_a}\right), \\ t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \\ t_w^* &= \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a), \end{aligned} \quad (8)$$

dimana x , y , w , dan h adalah koordinat tengah dari *bounding-box* disertai dengan tinggi dan lebarnya. x , x_a , dan x^* adalah *predicted box*, *anchor box*, dan *ground-truth box* (sama juga dengan y , w , h) [29].

2.6. Evaluation Metrics

2.6.1. Intersection over Union (IoU)



Gambar 2. 15 Ilustrasi IoU

IoU (*Intersection over Union*) atau biasa disebut dengan *Jaccard Index* adalah *metric* yang sering dipakai untuk membandingkan kemiripan (*similarity*) dari dua bentuk (*shape*). IoU mengkodekan bentuk dari sebuah objek pada perbandingan seperti panjang, lebar, ataupun lokasi dari *bounding boxes* pada sebuah *region* (daerah) dan kemudian menghitung langkah normalisasi yang berfokus pada area tersebut (dari *region*), yang membuat IoU *scale invariant*.

Dikarenakan ini juga, semua perhitungan performa untuk evaluasi dari segmentasi, deteksi objek, dan *tracking* bergantung pada *metric* ini.

IoU (*Intersection over Union*) adalah *metric* evaluasi untuk deteksi objek. Ini digunakan untuk menghitung *true positive* dan *false positive* pada satu set prediksi. Ketika menggunakan IoU sebagai *metric* evaluasi sebuah *threshold* harus dipilih. Seperti contoh *mean Average Precision* (mAP) dihitung berdasarkan IoU *threshold* seperti 0.5, namun pilihan IoU bukan berarti langsung sepenuhnya mempengaruhi performa dari lokalisasi dari metode yang berbeda-beda. IoU (*Intersection over Union*) melakukan perbandingan kemiripan antara dua bentuk (*shape*) dengan cara [37], [38]:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (9)$$

2.6.2. *mean Average Precision* (mAP)

True positive (TP) adalah deteksi yang benar dari *ground-truth bounding-box*. *False positive (FP)* adalah deteksi yang salah dari objek yang tidak ada atau deteksi yang salah tempat. *False negative (FN)* adalah *ground-truth bounding-box* yang tidak terdeteksi. *True negative (TN)* tidak dihitung karena kemungkinan hasilnya sangatlah banyak [38].

Precision adalah cara dari sebuah model untuk mengidentifikasi objek yang hanya relevan (persen prediksi positif yang benar). *Recall* adalah cara dari sebuah model untuk menemukan semua kasus yang relevan (persen prediksi positif dari semua *ground-truth*).

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{semua deteksi}} \quad (10)$$

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{semua ground - truth}} \quad (11)$$

mean AP (mAP) adalah *metric* yang dipakai untuk mengukur akurasi dari *object detector* di seluruh kelas. *mAP* adalah rata-rata AP dari seluruh kelas.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (12)$$

Dengan AP_i adalah AP dari kelas ke- i dan N adalah total kelas [34]. *AP* adalah:

$$AP = \frac{1}{11} \sum_{Recall_i} Precision(Recall_i) \quad (13)$$

2.7. Penelitian Terkait

Pada tahun 2020 Saiz, dkk. [39] melakukan penelitian dengan tujuan untuk mendeteksi virus Covid-19 secara cepat dengan menggunakan data dari citra *x-ray* dada pada paru-paru. Penelitian tersebut menggunakan 1500 data citra dari pasien yang tidak terinfeksi, pasien yang terinfeksi Covid-19 dan pasien yang terinfeksi pneumonia dengan tujuan untuk klasifikasi pasien dengan status positif ataupun negatif Covid-19. Penelitian tersebut berhasil memperoleh 94.92% *sensibility* dan 92.00% spesifitas pada deteksi Covid-19 dengan menggunakan model SSD300. Pada penelitian lain di tahun 2021 oleh Abbas, dkk [40] melakukan penelitian dengan tujuan klasifikasi Covid-19 menggunakan data dari citra *x-ray* dada. Dengan menggunakan DeTraC (*Decompose, Transfer, and Compose*), penelitian tersebut mampu untuk menghasilkan 93.1% akurasi, 100% *sensibility* dengan menggunakan data yang diperoleh dari berbagai macam rumah sakit yang ada di dunia.

Tabel 1 Penelitian terkait

Penelitian, Tahun	Permasalahan	Metode/Solusi	Hasil
Saiz, dkk. [39] 2020	Membuat model sistem deteksi objek dengan 1500 data dari pasien normal, Covid-19 dan pneumonia	Menggunakan model SSD300 serta dengan mengatur kontras dari data citra menggunakan teknik CLAHE (<i>Contrast Limited Adaptive Histogram Equalization</i>) dikarenakan faktor radiasi yang berbeda juga dibeberapa tempat <i>x-ray</i> .	Dengan menggunakan CLAHE didapatkan akurasi 92% pada kelas Covid-19 dan 94.92% pada kelas normal dibandingkan dengan ketika tidak menggunakannya (83% dan 93.24% untuk Covid-19 dan normal)

Abbas, dkk. [40] 2021	Membuat model dengan menerapkan konsep <i>transfer learning</i> dari <i>dataset</i> yang berjumlah 80 sampel normal, 105 Covid-19 dan 11 SARS	Menggunakan metode yang bernama DeTraC (<i>Decompose, Transfer, and Compose</i>)	Hasil dari penelitian tersebut menunjukkan bahwa DeTraC mendapatkan hasil 87.9% sensitifitas dan 100% spesifitas
Hasan, dkk [41] 2021	Membuat model untuk deteksi pneumonia dengan menggabungkan beberapa fitur dari <i>deep learning</i> Pada data yang berjumlah 7720	Membuat model dengan menggabungkan <i>Image data generator</i> , Model VGG19, AveragePooling2D, Flatten, <i>Dense layer</i> , dan <i>Dropout</i> .	Dengan <i>splitting</i> data sebesar 80% <i>training</i> dan 20% <i>testing</i> didapatkan hasil rata-rata akurasi sebesar 91.69%, sensitivitas sebesar 95.29% dan spesifitas sebesar 100%
Hemdan, dkk [42] 2020	Membuat model sistem klasifikasi Covid-19 dengan keterbatasan dalam <i>dataset</i> yang hanya berjumlah 50, dengan 25 data terkonfirmasi Covid-19 dan sisanya negatif	Membuat <i>framework deep learning</i> baru dengan nama COVIDX-Net yang terdiri dari 7 <i>arsitektur deep convolutional neural network</i> yaitu VGG19, DenseNet201, ResNetV2, Inception V3, InceptionResNetV2, Xception, MobileNetV2. <i>Dataset</i> di split dengan ketentuan 80% untuk <i>training</i> dan 20% untuk <i>testing</i> . Lalu	Tingkat akurasi mencapai 90% didapat oleh VGG19 dan DenseNet201 dengan presisi Covid mencapai 83%. Akurasi mencapai 80% didapat oleh InceptionResNetV2 dan Xception. Sedangkan ResNetV2 dan MobileNetV2 memperoleh akurasi masing-masing 70% dan 60%. Hasil paling rendah didapat oleh InceptionV3

		dilakukan tuning terhadap seluruh arsitektur secara bermasaan, dan hasil dari tiap arsitektur di evaluasi dengan confusion metrix	dengan hanya akurasi mencapai 50%
Gordienko, dkk [43] 2019	Menggunakan teknik <i>bone shadow exlusion</i> untuk <i>deep learning</i> segmentasi paru-paru pada citra <i>x-ray</i> dada	Menggunakan teknik <i>preprocssing bone elimination</i> untuk menghapus tulang pada citra yang kemudian hasil datanya akan dimasukkan kedalam UNet	Hasil yang didapat adalah dengan menggunakan citra yang sudah di <i>preprocessing</i> dengan menggunakan <i>bone elimination</i> didapatkan akurasi yang lebih bagus dibandingkan tanpa <i>preprocessing</i> .
Saad, dkk [44] 2014	Menggunakan <i>filter</i> untuk segmentasi paru-paru pada citra <i>x-ray</i> dada.	Menggunakan metode <i>euler number</i> yang akan menjadikan citra berbentuk biner untuk ekstraksi area paru-paru sebelum melakukan deteksi objek	Menggunakan evaluasi <i>jaccard similarity</i> didapatkan kemiripan rata-rata yaitu 0.7
Dai W, dkk [45] 2017	Membuat sistem segmentasi organ jantung dan paru-paru dengan menggunakan citra <i>x-ray</i> dada.	Menggunakan SCAN (<i>Structure Correcting Adversarial Network</i>) untuk membedakan <i>ground-truth</i> dari <i>mask</i> dengan <i>segmentation network (FCN)</i> untuk mendapatkan struktur citra yang lebih penting.	Dengan menggunakan SCAN pada 209 & 117 <i>dataset</i> dan 38 & 21 <i>dataset</i> validasi secara acak (dari JSRT dan Montgomery) didapatkan IoU yang lebih tinggi yaitu 95.1% untuk <i>dataset</i> JSRT dan 93.0% untuk

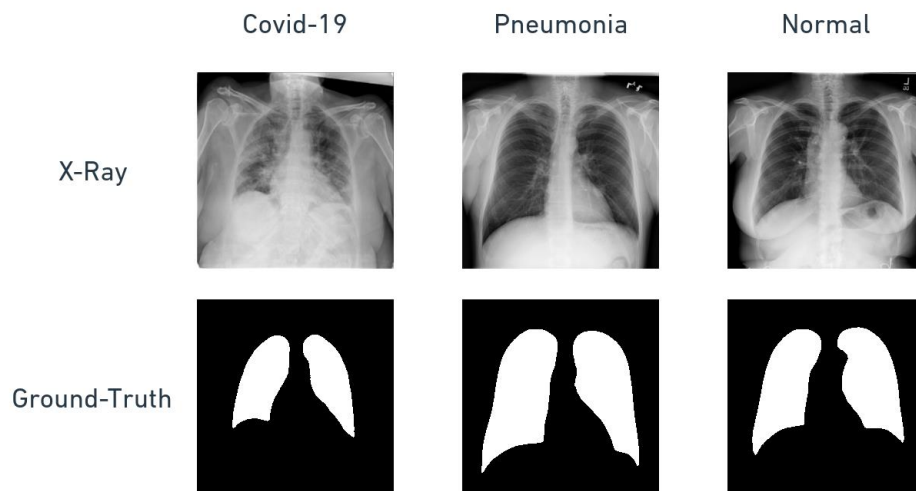
			<i>dataset</i> Montgomery
Munawar, dkk [46] 2020	Membuat model untuk segmentasi paru-paru menggunakan citra <i>x-ray</i> dada.	Menggunakan <i>Generative Adversarial Networks</i> (GAN) untuk segmentasi paru-paru pada citra <i>x-ray</i> dada sebisa mungkin untuk mendapatkan hasil yang paling realistis	Dengan menggunakan GAN pada <i>dataset</i> CXR yang berbeda (JSRT, Montgomery, <i>Shenzen Chest X-Ray</i>) didapatkan hasil berupa <i>dice-score</i> sebesar 0.9740 dan IoU sebesar 0.943
Chouhan, dkk [47] 2020	Deteksi pneumonia menggunakan <i>deep learning</i> dan mencari model yang terbaik menggunakan <i>transfer learning</i>	Menggunakan model <i>deep learning</i> AlexNet, DenseNet121, InceptionV3, ResNet18, dan GoogLeNet dan <i>pretrained</i> model dari AlexNet, DenseNet121, InceptionV3, ResNet18, dan GoogLeNet dengan menggunakan 5232 <i>dataset</i> dimana 1346 mempunyai kelas normal, 3883 mempunyai kelas Pneumonia yang terdiri dari 2538 kelas bakteri pneumonia, dan 1345 kelas virus pneumonia	Model terbaik berhasil mendapatkan hasil berupa akurasi sebesar 96.4% dan <i>recall</i> sebesar 99.62% dengan menggunakan data dari <i>Guangzhou Women and Children's Medical Center</i>
Al-Waisy, dkk [48] 2021	Menggunakan citra <i>x-ray</i> dada untuk identifikasi Covid-19 dengan	Menggunakan <i>deep learning</i> COVID-DeepNet untuk deteksi pasien	COVID-DeepNet bisa mendeteksi pasien dengan Covid-19 dengan

	<p>menggunakan <i>deep learning</i> yang bernama COVID-DeepNet</p>	<p>Covid-19 yang menggunakan arsitektur paralel pada data <i>x-ray</i> dad sebanyak 400 <i>dataset</i> dengan kelas Covid-19 dari repositori Github Cohen, Radiopaedia, <i>Italian Society of Medical and Interventional Radiology</i> (SIRM), dan <i>Radiological Society of North America</i> (RSNA). Dan juga <i>dataset</i> normal berjumlah 400 dari <i>Kaggle's CX-R image</i> (Pneumonia)</p>	<p>akurasi sebesar 99.39%, sensitifitas sebesar 99.90%, spesifitas sebesar 100%, presisi sebesar 100%, <i>F1-score</i> sebesar 99.93%, MSE sebesar 0.021% dan RMSE sebesar 0.016%</p>
--	--	--	---

BAB III METODE USULAN

3.1. Dataset

Dataset citra *x-ray* dada yang digunakan dalam penelitian ini adalah *dataset* publik bersumber dari Qatar University yang tersedia dalam repositori Kaggle dengan situs: <https://www.kaggle.com/datasets/anasmohammedtahir/covidqu> . *Dataset* memiliki ukuran 256x256 piksel dan berformat .png yang terdiri dari 3 buah kategori yakni paru-paru terindikasi Covid-19, Pneumonia, dan paru-paru normal. Berikut sampel dari citra *x-ray* beserta dengan *ground-truth*-nya:



Gambar 3. 1 Contoh data *x-ray* beserta *ground-truth*

Jumlah data citra *x-ray* beserta *ground-truth* terdapat 33.920 data dengan 11.956 data citra yang terinfeksi Covid-19, 11.263 terinfeksi Pneumonia, dan 10.701 citra dengan paru-paru normal. Penelitian ini hanya akan menggunakan 1500 data (diambil secara acak menggunakan program) dengan setiap kelasnya terdiri dari 500 data. Untuk anotasi data bisa dilakukan dengan menggunakan program.

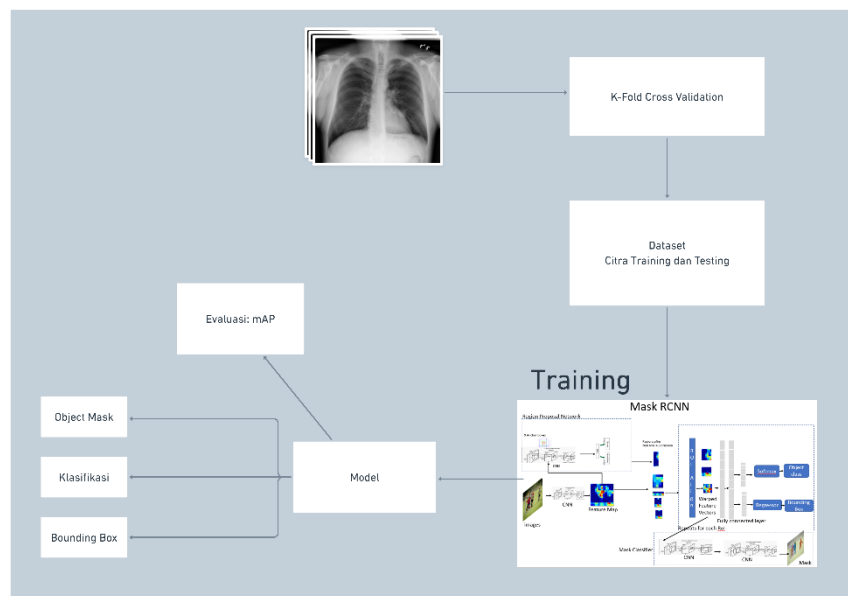
Untuk mengevaluasi model dari *Mask R-CNN*, akan dilakukan *K-Fold Cross Validation* dengan $K = 5$. *Fold* pertama akan dipakai untuk *testing* model sedangkan *Fold* sisanya akan dipakai untuk *training*.

Tabel 2 Pembagian data

Data Training(K-Fold)	Data Testing
-----------------------	--------------

Jumlah <i>Training Set</i>	1080	Kategori	Jumlah
		Covid-19	50
Jumlah <i>Validation Set</i>	270	Pneumonia	50
		Normal	50
Total Data	1350	Jumlah	150

3.2. Arsitektur Diagram *Training* dan *Testing*



Gambar 3. 2 Diagram *Training* dan *Testing*

Proses pada gambar 3.2 adalah sebagai berikut:

1. Pertama, *dataset* akan dilakukan *cross-validation* dengan menggunakan k-fold dengan nilai k adalah 5
2. Kemudian akan dilakukan *training* secara *end-to-end* dan didapatkan hasil yaitu model yang sudah terlatih.
3. Pada proses *training*, dilakukan beberapa hal yaitu sebagai berikut:
 - a. Memasukkan citra input
 - b. Citra akan diproses oleh arsitektur *Mask R-CNN* untuk mendapatkan *bounding box*, kelas, dan juga hasil segmentasi.
 - c. Kemudian akan dihitung mAP dari citra *testing* dan hasil prediksi

3.3. Skenario Uji Coba

Skenario uji coba akan dilakukan untuk mencari model dengan performa yang paling bagus dan juga kinerja terbaik. Pertama akan dilakukan *cross validation* yaitu membagi data menjadi beberapa bagian yang akan diuji setiap bagiannya menggunakan k yang bernilai 5. Data yang dibagi-bagi tadi akan secara bergantian menjadi data *testing* dan juga data *training*.

Selanjutnya akan dilakukan *training Mask R-CNN* dengan data yang sudah dibagi tadi menjadi beberapa *fold*. Disamping itu akan digunakan juga pergantian *threshold* dari IoU dengan pergantian 0.4, 0.5, 0.6 dan 0.7. Dan pada tahap terakhir akan dilakukan evaluasi menggunakan *mean Average Precision* (mAP). Tujuan dari pengukuran tersebut adalah untuk mengetahui performa dan presisi dari sebuah model untuk deteksi yang dilakukan pada tiap *fold*.

3.4. Perkiraan Jadwal

Tabel 3 Tabel perkiraan jadwal

No	Kegiatan	Bulan 1				Bulan 2				Bulan 3				Bulan 4				Bulan 5				Bulan 6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Mempelajari materi <i>Machine Learning</i> dan <i>Deep Learning</i>																								
2	Mempelajari <i>instance segmentation</i>																								
3	Mencari referensi penelitian sebelumnya tentang <i>instance segmentation</i> paru-paru pada citra <i>x-ray</i> dada																								
4	Mengumpulkan dataset citra <i>x-ray</i> dada																								
5	Perancangan sistem																								
6	Penyusunan proposal																								
7	Mengacak data																								
8	Membagi data <i>training</i> dan <i>testing</i>																								
9	Membangun program untuk <i>import</i> data																								
10	Membangun program untuk pelatihan data																								
11	Proses pelatihan data																								
12	Membangun program untuk pengujian model																								
13	Proses pengujian data																								
14	Uji coba keseluruhan sistem																								
15	Evaluasi																								
16	Penyusunan laporan																								
17	Sidang akhir																								
18	Dokumentasi dan studi literatur																								

DAFTAR PUSTAKA

- [1] M. Ciotti *et al.*, “COVID-19 Outbreak: An Overview,” *Chemotherapy*, vol. 64, no. 5–6, pp. 215–223, 2019, doi: 10.1159/000507423.
- [2] WHO, “WHO Coronavirus (COVID-19) Dashboard,” <https://covid19.who.int/>, Dec. 08, 2022.
- [3] WHO, “Coronavirus disease (COVID-19),” <https://www.who.int/westernpacific/health-topics/detail/coronavirus>, Dec. 08, 2022.
- [4] B. Giri, S. Pandey, R. Shrestha, K. Pokharel, F. S. Ligler, and B. B. Neupane, “Review of analytical performance of COVID-19 detection methods,” *Anal Bioanal Chem*, vol. 413, no. 1, pp. 35–48, Jan. 2021, doi: 10.1007/s00216-020-02889-x.
- [5] L. Falzone, G. Gattuso, A. Tsatsakis, D. Spandidos, and M. Libra, “Current and innovative methods for the diagnosis of COVID-19 infection (Review),” *Int J Mol Med*, vol. 47, no. 6, p. 100, Apr. 2021, doi: 10.3892/ijmm.2021.4933.
- [6] M. Mahmud, M. S. Kaiser, T. M. McGinnity, and A. Hussain, “Deep Learning in Mining Biological Data,” *Cognit Comput*, vol. 13, no. 1, pp. 1–33, Jan. 2021, doi: 10.1007/s12559-020-09773-x.
- [7] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, “Applications of Deep Learning and Reinforcement Learning to Biological Data,” Nov. 2017, doi: 10.1109/TNNLS.2018.2790388.
- [8] I. W. Harsono, S. Liawatimena, and T. W. Cenggoro, “Lung nodule detection and classification from Thorax CT-scan using RetinaNet with transfer learning,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 567–577, Mar. 2022, doi: 10.1016/j.jksuci.2020.03.013.
- [9] L. Wang, Z. Q. Lin, and A. Wong, “COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images,” *Sci Rep*, vol. 10, no. 1, p. 19549, Nov. 2020, doi: 10.1038/s41598-020-76550-z.

- [10] L. Gaur, U. Bhatia, N. Z. Jhanjhi, G. Muhammad, and M. Masud, "Medical image-based detection of COVID-19 using Deep Convolution Neural Networks," *Multimed Syst*, Apr. 2021, doi: 10.1007/s00530-021-00794-6.
- [11] S. Podder, S. Bhattacharjee, and A. Roy, "An efficient method of detection of COVID-19 using Mask R-CNN on chest X-Ray images," *AIMS Biophys*, vol. 8, no. 3, pp. 281–290, 2021, doi: 10.3934/biophy.2021022.
- [12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," Mar. 2017.
- [13] Medlineplus, "X-Rays: Medline Plus."
- [14] St Vincent's Lung Health, "Chest X ray - St Vincent's Lung Health."
- [15] U.S. Department of Health & Human Services, "X-rays."
- [16] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [17] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 1271–1278. doi: 10.1109/CVPR.2009.5206532.
- [18] M. Sonka and J. M. Fitzpatrick, "Handbook of Medical Imaging, Volume 2: Medical Image Processing and Analysis".
- [19] R. Sarma and Y. K. Gupta, "A comparative study of new and existing segmentation techniques," *IOP Conf Ser Mater Sci Eng*, vol. 1022, no. 1, p. 012027, Jan. 2021, doi: 10.1088/1757-899X/1022/1/012027.
- [20] A. Arnab and P. H. S. Torr, "Pixelwise Instance Segmentation with a Dynamically Instantiated Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 879–888. doi: 10.1109/CVPR.2017.100.
- [21] S. Gonzalez, C. Arellano, and J. E. Tapia, "Deepblueberry: Quantification of Blueberries in the Wild Using Instance Segmentation," *IEEE Access*, vol. 7, pp. 105776–105788, 2019, doi: 10.1109/ACCESS.2019.2933062.
- [22] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN,"

- Comput Electron Agric*, vol. 163, p. 104846, Aug. 2019, doi: 10.1016/j.compag.2019.06.001.
- [23] J. Wu, *Introduction to Convolutional Neural Networks*. 2017.
 - [24] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," Nov. 2015.
 - [25] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.
 - [26] S. Albawi, O. Bayat, S. Al-Azawi, and O. N. Ucan, "Social Touch Gesture Recognition Using Convolutional Neural Network," *Comput Intell Neurosci*, vol. 2018, pp. 1–10, Oct. 2018, doi: 10.1155/2018/6973103.
 - [27] H. H. Sultan, N. M. Salem, and W. Al-Atabany, "Multi-Classification of Brain Tumor Images Using Deep Neural Network," *IEEE Access*, vol. 7, pp. 69215–69225, 2019, doi: 10.1109/ACCESS.2019.2919122.
 - [28] T. Leibovich-Raveh, D. J. Lewis, S. Al-Rubaiey Kadhim, and D. Ansari, "A new method for calculating individual subitizing ranges," *J Numer Cogn*, vol. 4, no. 2, pp. 429–447, Sep. 2018, doi: 10.5964/jnc.v4i2.74.
 - [29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jun. 2015.
 - [30] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," May 2017.
 - [31] J. Hosang, R. Benenson, and B. Schiele, "A convnet for non-maximum suppression," Nov. 2015.
 - [32] W. T. Vetterling and B. P. Flannery, *Numerical recipes in C: the art of scientific computing*. New York: Cambridge University Press, 1992.
 - [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015.
 - [34] Y. Tong, W. Lu, Q. Deng, C. Chen, and Y. Shen, "Automated identification of retinopathy of prematurity by image-based deep learning," *Eye and Vision*, vol. 7, no. 1, p. 40, Dec. 2020, doi: 10.1186/s40662-020-00206-2.

- [35] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 4, pp. 640–651, Apr. 2017, doi: 10.1109/TPAMI.2016.2572683.
- [36] R. Girshick, "Fast R-CNN," Apr. 2015.
- [37] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," Feb. 2019.
- [38] "A Survey on Performance Metrics for Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*.
- [39] F. Saiz and I. Barandiaran, "COVID-19 Detection in Chest X-ray Images using a Deep Learning Approach," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 2, p. 4, 2020, doi: 10.9781/ijimai.2020.04.003.
- [40] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Applied Intelligence*, vol. 51, no. 2, pp. 854–864, Feb. 2021, doi: 10.1007/s10489-020-01829-7.
- [41] M. D. K. Hasan *et al.*, "Deep Learning Approaches for Detecting Pneumonia in COVID-19 Patients by Analyzing Chest X-Ray Images," *Math Probl Eng*, vol. 2021, pp. 1–8, May 2021, doi: 10.1155/2021/9929274.
- [42] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images," Mar. 2020.
- [43] Yu. Gordienko *et al.*, "Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer," 2019, pp. 638–647. doi: 10.1007/978-3-319-91008-6_63.
- [44] M. N. Saad, Z. Muda, N. S. Ashaari, and H. A. Hamid, "Image segmentation for lung region in chest X-ray images using edge detection and morphology," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, Nov. 2014, pp. 46–51. doi: 10.1109/ICCSCE.2014.7072687.

- [45] W. Dai *et al.*, “SCAN: Structure Correcting Adversarial Network for Organ Segmentation in Chest X-rays,” Mar. 2017.
- [46] F. Munawar, S. Azmat, T. Iqbal, C. Gronlund, and H. Ali, “Segmentation of Lungs in Chest X-Ray Image Using Generative Adversarial Networks,” *IEEE Access*, vol. 8, pp. 153535–153545, 2020, doi: 10.1109/ACCESS.2020.3017915.
- [47] V. Chouhan *et al.*, “A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images,” *Applied Sciences*, vol. 10, no. 2, p. 559, Jan. 2020, doi: 10.3390/app10020559.
- [48] A. S. Al-Waisy *et al.*, “COVID-DeepNet: Hybrid Multimodal Deep Learning System for Improving COVID-19 Pneumonia Detection in Chest X-ray Images,” *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2409–2429, 2021, doi: 10.32604/cmc.2021.012955.