

Segmentasi Citra X-ray Paru dengan Deep Learning

Muhammad Hussein^{*1}, Agus Eko Minarno², Yufis Azhar³

^{1,2,3}Universitas Muhammadiyah Malang

hussein.muhammad47@gmail.com^{*1}, aguseko@umm.ac.id², yufis.az@gmail.com³

Abstrak

Segmentasi gambar merupakan salah satu hal utama dalam kajian computer vision dan image processing. Salah satu contohnya adalah pemrosesan gambar x-ray paru untuk mengetahui penyakit-penyakit yang ada di dalam paru. U-Net merupakan salah satu model segmentasi yang telah dibuat untuk mempermudah seseorang membangun model untuk segmentasi gambar. U-net bisa digunakan pada gambar apapun. Dari keunggulannya itu peneliti mencoba menggunakan U-Net dikombinasikan dengan Inception, MobileNet dan EfficientNet untuk melakukan segmentasi pada gambar medis x-ray paru. Gambar di resize menjadi 512 x 512 pixel. Augmentasi yang dilakukan adalah zoom range, height shift, width shift dan horizontal flip. Epoch sebanyak 200 dan batch size sebesar 4. Skenario terbaik pada penelitian ini adalah dengan menggunakan U-net Efficientnetb0 dengan nilai dice 0.967 dan Jaccard 0.937.

Kata Kunci: Segmentasi, Deep Learning, CNN, X-ray, Paru-paru

Abstract

Image segmentation is one of the main things in the study of computer vision and image processing. One example is the processing of lung x-ray images to find out diseases in the lungs. U-net is a segmentation model that has been created to make it easier for someone to build a model for image segmentation. U-net can be used on any image. From its advantages, the researchers tried to use U-net in combination with Inception, MobileNet and EfficientNet to segment medical x-ray images of the lungs. The image is resized to 512 x 512 pixels. Augmentation that is done is zoom range, height shift, width shift and horizontal flip. Epoch is 200 and batch size is 4. The best scenario in this research is to use U-net Efficientnetb0 with dice value of 0.967, Jaccard of 0.937.

Keywords: Segmentation, Deep Learning, CNN, X-Ray, Lung

1. Pendahuluan

Kemajuan teknologi komputer membuat teknik pemrosesan gambar menjadi semakin penting dalam berbagai aplikasi. Salah satu teknik pemrosesan gambar adalah metode segmentasi. Segmentasi gambar merupakan subjek klasik dalam pengolahan citra dan juga merupakan titik pusat dan fokus pada teknik pengolahan gambar (1). Segmentasi gambar mengacu pada proses untuk membagi gambar menjadi beberapa wilayah sesuai dengan karakteristik visual yang dimiliki oleh piksel gambar. Implementasi segmentasi dengan aplikasi penting seperti pemahaman adegan, analisis citra medis, persepsi robot, pengawasan video, *augmented reality*, dan kompresi gambar (2). Dari perspektif yang lebih teknis, segmentasi gambar merupakan proses pelabelan setiap piksel dalam gambar sehingga piksel dengan label yang sama terhubung satu sama lain dalam beberapa properti visual atau semantik (3).

Dalam hal medis, metode segmentasi memegang peran sangat penting. Banyak peneliti mengajukan teknik segmentasi yang berbeda namun belum menyarankan metode umum yang dapat diterapkan karena setiap hasil dari gambar medis memiliki karakteristik yang berbeda-beda. Beberapa penerapan teknik terbaru yang digunakan dalam metode segmentasi gambar medis, yaitu jaringan saraf tiruan (*artificial neural network*), model bentuk aktif (*active shape model*), kecerdasan buatan (*artificial intelligence*), dan pengelompokan (*Clustering*) (4).

Paru-paru merupakan organ dalam tubuh manusia yang digunakan untuk proses pertukaran antara oksigen dan karbon dioksida. Proses tersebut terjadi pada bagian paru yang disebut alveolus. Jika terjadi masalah dengan paru-paru maka proses bernafas pun akan terganggu. Penyakit paru-paru yang banyak diderita oleh masyarakat diantaranya Penyakit Paru Obstruktif Kronik (PPOK), tuberkulosis paru, kanker paru, asma bronkial, dan pneumonia (5).

Beberapa penyakit tersebut akan menyebabkan perubahan bentuk pada paru-paru jika dilihat dari gambar medis yang dihasilkan seperti *Magnetic Resonance Imaging* (MRI) atau *Computed Tomography Scan* (CT Scan)

Beberapa penelitian tentang segmentasi dengan jaringan saraf tiruan yaitu pertama penelitian yang dilakukan oleh Ronneberger et al. pada tahun 2015 mengusulkan arsitektur yang diberi nama "*U-net*" (6). Arsitektur ini merupakan modifikasi dari model *Fully Convolutional Network* dari penelitian yang dilakukan oleh Jonathan Long et al. (7). Modifikasi yang dilakukan di bagian *upsampling* yaitu dengan mengubah saluran fitur sehingga jaringan bisa mengirimkan konteks informasi ke lapisan resolusi yang lebih tinggi. Modifikasi ini secara tidak sengaja menghasilkan arsitektur *layer* seperti huruf *U*. Skenario pengujian dilakukan dengan menguji *U-net* di beberapa tugas segmentasi yang berbeda. Hasilnya dapat dilihat pada tabel 1 bahwa *U-net* memberikan performa terbaik dari beberapa arsitektur segmentasi lain dalam *ISBI cell tracking challenge 2015*.

Tabel 1. Hasil Segmentasi dalam *ISBI cell tracking challenge 2015*

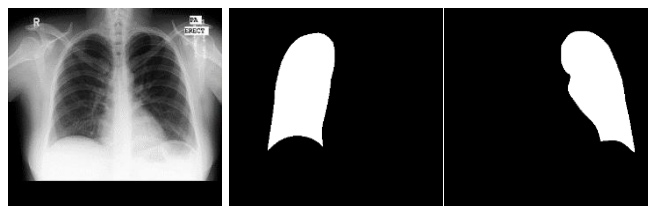
Nama Model	PhC - U373	DIC - HeLa
U-net (2015)	0.9203	0.7756
second-best 2015	0.83	0.46
HOUS-US (2014)	0.5323	-
KTH-SE (2014)	0.7953	0.4607
IMCB-SG (2014)	0.2669	0.2935

Kemudian penelitian yang dilakukan oleh Jyoti et al. pada tahun 2018 (8) mengusulkan pengembangan kerangka kerja otomatis untuk segmentasi paru-paru pada gambar *x-ray* toraks menggunakan Jaringan Syaraf Konvolusional Dalam (*Deep Convolutional Neural Network*) berdasarkan *U-net*. Dataset yang digunakan dalam penelitian ini dari Montgomery County, Maryland, dan Shenzhen People's Hospital di China. Dataset ini dikelola oleh National Library of Medicine (NLM), National Institutes of Health (NIH) (9). Skenario pada penelitian ini yaitu setiap *input* gambar di *resize* menjadi 512 x 512 piksels sebelum masuk ke model Deep CNN. Dataset dibagi menjadi beberapa bagian yaitu 80% untuk data latih dan 20% untuk data uji. Dari data latih 10% digunakan untuk data validasi. Optimizer menggunakan Adam dengan *learning rate* 0.0005, batch size sebesar 4 dan 200 epoch. Proses *augmentasi* yang dilakukan yaitu *zoom range* 0.05, *height shift*, *width shift* dan *horizontal flip*. Hasil dari penelitian ini di evaluasi dengan menggunakan *dice* dengan hasil 98.6. prediksi model segmentasi ini lebih baik sekitar 2.2% dari penelitian-penelitian yang sebelumnya. Model lain juga dibangun dengan menggunakan *pretrained model*, yaitu menerapkan *skip connection* atau ResNet, namun *skip connection* tidak membantu meningkatkan hasil dari segmentasi.

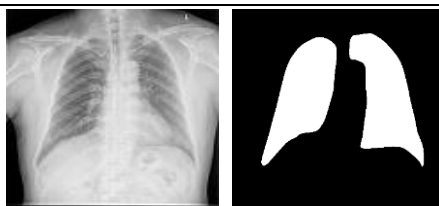
Berdasarkan kedua penelitian sebelumnya, peneliti akan melakukan penelitian segmentasi gambar *x-ray* toraks dengan menggunakan arsitektur *U-net* dan *encoder* menggunakan arsitektur *pretrained model* seperti *Inception*, *MobileNet* dan *EfficientNet*

2. Metode Penelitian

Dataset yang digunakan dalam penelitian ini yaitu dataset dari Montgomery County dan Shenzhen. Pada dataset dari Montgomery County ada 138 gambar posterior-anterior (80 normal dan 58 lainnya memiliki kelainan). Dataset dari Shenzhen memiliki 326 normal dan 336 abnormal. Dataset Montgomery sudah menyediakan masking manual dan untuk masking dari data Shenzhen disediakan oleh penelitian S. Stirenko et al (10). Jadi total dataset ada 800 gambar dan 800 *mask*.



Gambar 1. Contoh data dan manual masking dari data Montgomery



Gambar 2. Contoh data dan *masking* manual dari data Shenzhen

Dataset yang terdiri dari *x-ray* dan *masking* akan dibagi menjadi 80% data latih dan 20% data uji. Data validasi akan diambil dari 10% data latih.

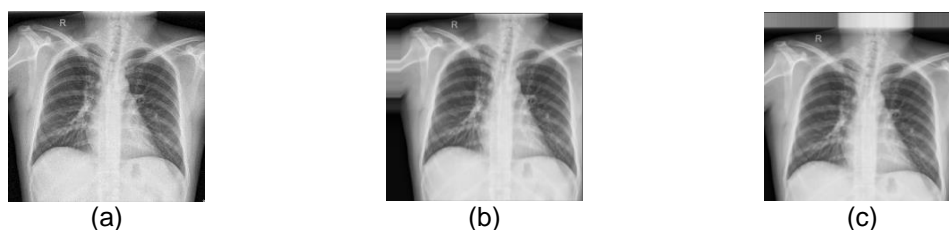
Preprocessing data dilakukan agar data yang diproses pada saat training menghasilkan keluaran yang baik. Proses preprocessing yang dilakukan mengikuti paper referensi yaitu semua gambar pada data latih, data validasi, dan data uji akan dilakukan proses *resize* menjadi ukuran 512 x 512 piksel.

Augmentasi data dilakukan untuk meningkatkan volume dari data latih. Proses augmentasi yang dilakukan akan mengikuti penelitian referensi yaitu *zoom range*, *height shift*, *width shift* dan *horizontal flip*.



Gambar 3. Augmentasi *zoom range*: (a) gambar asli; (b) gambar hasil *zoom range*

Shift (pergeseran) gambar berarti memindahkan semua piksel gambar ke satu arah, seperti horizontal atau vertikal, sambil menjaga dimensi gambar tetap sama. Ini berarti beberapa piksel akan terpotong dari gambar dan akan ada wilayah gambar di mana nilai piksel baru harus ditentukan. *Height shift* akan menggeser gambar secara vertikal dan *width shift* menggeser secara horizontal, dan jaraknya tergantung dari nilai yang ditentukan.



Gambar 4. Augmentasi *shift*: (a) gambar asli; (b) hasil *width shift*; (c) hasil *height shift*

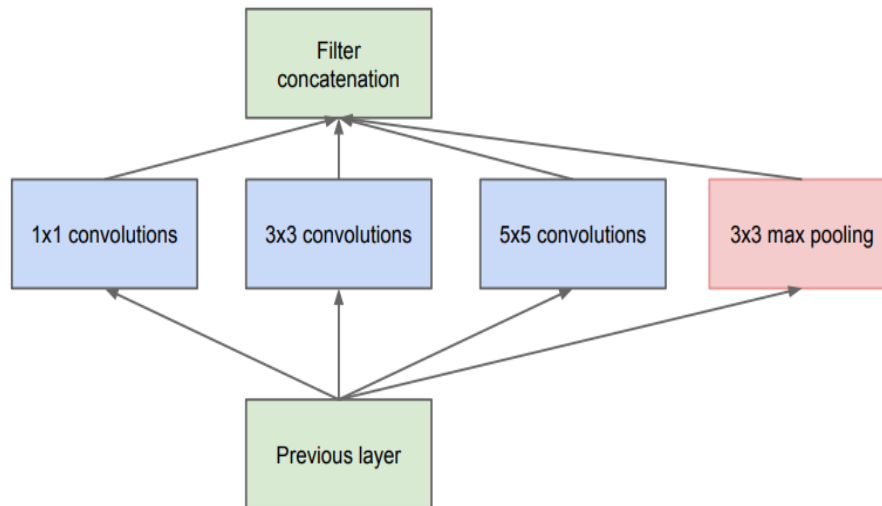


Gambar 5. Augmentasi *horizontal flip*: (a) gambar asli; (b) hasil *horizontal flip*

Model segmentasi yang digunakan dalam penelitian ini menggunakan *U-Net*. *U-net* akan digunakan dengan *pretrained model*. *Pretrained Model* adalah sebuah model pada *deep learning* yang sudah dilatih terlebih dahulu. Pada penelitian ini akan menggunakan beberapa pretrained model untuk meningkatkan hasil segmentasi. *Pretrained model* disini digunakan sebagai encoder dari *U-net*. Pretrained model yang akan digunakan adalah *Inception* (11), *MobileNet* (12) dan *EfficientNet* (13). Untuk membangun model tersebut pustaka *segmentation_model* (14)

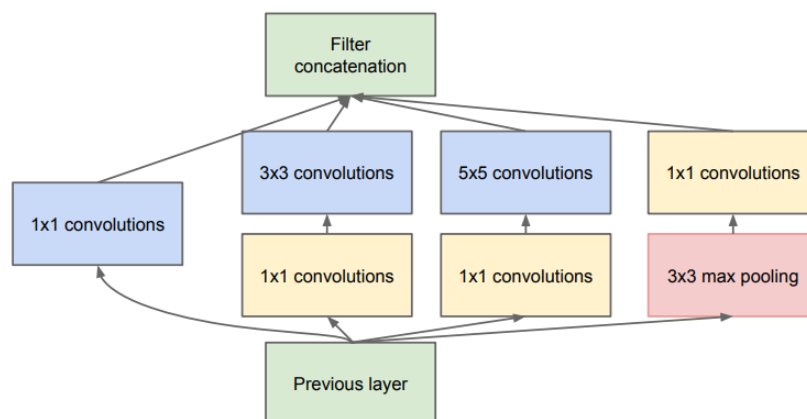
digunakan. Pustaka *segmentation_model* menyediakan beberapa model segmentasi seperti *U-net* (6), *FPN*, *Linknet* (15), *PSPNet* (16).

Poin utama dari arsitektur *Inception* didasarkan pada pencarian bagaimana struktur sparse lokal yang optimal dalam jaringan visi konvolusional dapat didekati dan dicakup oleh komponen *dense* yang tersedia. Pada *Naive inception module* dilakukan proses konvolusi masukan dengan 3 ukuran filter yaitu 1×1 , 3×3 , dan 5×5 . *Max pooling* juga dilakukan, sedangkan keluaran dikirim ke modul *inception* selanjutnya. Struktur layer *Naive inception* dapat dilihat pada Gambar 6.



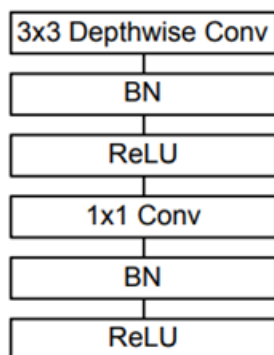
Gambar 6. Modul Inception versi Naive

Untuk mengatasi komputasi yang tidak efisien dari *Deep Neural Network*, jumlah *channel* masukan dibatasi dengan menambahkan konvolusi 1×1 sebelum konvolusi 3×3 dan konvolusi 5×5 , dan juga dapat diatasi dengan mengurangi jumlah masukan *channel*. Dibandingkan dengan modul *inception* versi *naive*, konvolusi 1×1 dimasukkan setelah *max pooling* layer. Modul *inception* dengan pengurangan dimensi bida dilihat pada Gambar 7.



Gambar 7. Modul Inception dengan pengurangan dimensi

Model *MobileNet* didasarkan pada *Deep Separable Convolution* yang merupakan bentuk konvolusi kumulatif yang memfaktorkan konvolusi standar menjadi konvolusi *depthwise* dan *pointwise*. Konvolusi *depthwise* menerapkan satu filter ke setiap *input channel*. Kemudian konvolusi *pointwise* menerapkan konvolusi 1×1 untuk menggabungkan *output* konvolusi *depthwise*. Konvolusi standar menyaring dan menggabungkan masukan menjadi satu set keluaran baru dalam satu langkah. Konvolusi *depthwise* membaginya menjadi dua lapisan, lapisan *filtering* dan lapisan *concatenating*. Faktorisasi ini berpengaruh secara signifikan dalam mengurangi ukuran perhitungan dan ukuran model.



Gambar 8. Konvolusi Separable Depthwise dengan lapisan Depthwise dan Pointwise diikuti oleh Batchnorm dan ReLU

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Gambar 9. Arsitektur MobileNet

EfficientNet adalah arsitektur CNN (*Convolutional Neural Network*) dan metode penskalaan yang secara seragam menskalakan semua dimensi jaringan dari kedalaman (*depth*), lebar (*width*), resolusi (*resolution*) menggunakan koefisien gabungan (ϕ). Persamaan penskalaan ini bisa dilihat pada Persamaan 1 berikut.

$$\begin{aligned}
 \text{depth: } d &= \alpha^\phi \\
 \text{width: } w &= \beta^\phi \\
 \text{resolution: } r &= \gamma^\phi \\
 \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\
 \alpha \geq 1, \beta \geq 1, \gamma \geq 1
 \end{aligned} \tag{1}$$

α , β , γ adalah konstanta yang bisa ditentukan menggunakan *grid search* kecil. ϕ adalah koefisien yang ditentukan pengguna yang mengontrol jumlah sumber daya yang tersedia untuk menskalakan model, sementara α , β , γ menentukan bagaimana sumber daya tambahan tersebut harus dibagi untuk lebar, kedalaman, dan resolusi masing-masing jaringan.

Jaringan dasar *EfficientNet* didasarkan pada blok *residual bottleneck* terbalik dari *MobileNetV2* (17). Arsitektur *EfficientNet* bisa dilihat pada Gambar 10.

Stage i	Operator \mathcal{F}_i	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	224 × 224	32	1
2	MBCConv1, k3x3	112 × 112	16	1
3	MBCConv6, k3x3	112 × 112	24	2
4	MBCConv6, k5x5	56 × 56	40	2
5	MBCConv6, k3x3	28 × 28	80	3
6	MBCConv6, k5x5	14 × 14	112	3
7	MBCConv6, k5x5	14 × 14	192	4
8	MBCConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

Gambar 10. Arsitektur EfficientNet

Pelatihan model akan menggunakan tiga skenario. Skenario pertama menggunakan *U-net Inception*, skenario kedua menggunakan *U-net MobileNet* dan skenario ketiga menggunakan *U-net EfficientNet*. Adapun untuk model dari jurnal referensi juga dilatih untuk mengetahui performa model dalam lingkungan yang sama dengan tiga model yang dibangun sebelumnya.

Pada proses pelatihan model, *epoch* yang digunakan sebanyak 200 dan *batch size* sebesar 4. *Optimizer* yang digunakan *Adam* dengan *learning rate* sebesar 0.0005. Semua hasil output dari satu skenario akan disimpan untuk membandingkan skenario mana yang terbaik. Mulai dari lama waktu training, matrik evaluasi dalam pelatihan dan model dengan *loss* paling kecil.

Metode evaluasi yang digunakan adalah uji segmentasi. Uji segmentasi ini merupakan tahapan yang dilakukan untuk mengukur nilai performa dari model yang telah dibangun. Pengukuran kinerja model dilakukan dengan menghitung nilai *dice* (18) dan nilai *Jaccard* (19). Persamaan untuk *dice* bisa dilihat pada Persamaan 2.

$$DICE = \frac{|S \cap GT|}{|S| + |GT|} = \frac{2|TP|}{2|TP| + |FN| + |FP|} \quad (2)$$

Dice koefisien adalah tumpang tindih antara *ground truth* (GT), dan hasil prediksi model tersegmentasi yang dihitung (S). Model dengan nilai *dice* yang besar menunjukkan performa segmentasi yang bagus.

Persamaan untuk *jaccard* bisa dilihat pada Persamaan 3. *Jaccard* membandingkan anggota untuk dua set untuk melihat anggota mana yang dibagi dan yang berbeda. Semakin tinggi.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3)$$

3. Hasil Penelitian dan Pembahasan

Penelitian ini menggunakan tiga skenario. Skenario pertama menggunakan *U-net Inception*, skenario kedua menggunakan *U-net MobileNet* dan skenario ketiga menggunakan *U-net EfficientNet*. Adapun untuk pelatihan model dari jurnal rujukan juga dilakukan. Pelatihan model jurnal rujukan dilakukan untuk mengetahui performa model tersebut pada lingkungan yang sama dengan model yang telah dibangun sebelumnya. Rasio dataset, *preprocessing*, parameter pelatihan dan metode evaluasi mengikuti dari jurnal rujukan. Hasil dari setiap pelatihan yang sudah disimpan akan dimuat kembali untuk melakukan perbandingan model.

Pelatihan model pertama dengan *U-net InceptionV3* memakan waktu sebanyak 250.25 menit. Pada data latih nilai *loss* yang diperoleh sebesar 0.029, nilai *dice* sebesar 0.971 dan nilai *jaccard* sebesar 0.944. Pada data validasi nilai *loss* yang diperoleh sebesar 0.032, nilai *dice* sebesar 0.968 dan nilai *jaccard* sebesar 0.938. Pada data uji nilai *dice* yang diperoleh sebesar 0.964 dan nilai *jaccard* sebesar 0.931

Pelatihan model kedua dengan *U-net Mobilenet* memakan waktu sebanyak 246.68 menit. Pada data latih nilai *loss* yang diperoleh sebesar 0.03, nilai *dice* sebesar 0.970 dan nilai *jaccard* sebesar 0.943. Pada data validasi nilai *loss* yang diperoleh sebesar 0.032, nilai *dice* sebesar

0.969 dan nilai *jaccard* sebesar 0.940. Pada data uji nilai *dice* yang diperoleh sebesar 0.964 dan nilai *jaccard* sebesar 0.932.

Pelatihan model ketiga dengan *U-net Efficientnetb0* memakan waktu sebanyak 251.64 menit. Pada data latih nilai *loss* yang diperoleh sebesar 0.028, nilai *dice* sebesar 0.972 dan nilai *jaccard* sebesar 0.946. Pada data validasi nilai *loss* yang diperoleh sebesar 0.031, nilai *dice* sebesar 0.969 dan nilai *jaccard* sebesar 0.940. Pada data uji nilai *dice* yang diperoleh sebesar 0.967 dan nilai *jaccard* sebesar 0.937.

Pelatihan model keempat dengan model dari jurnal rujukan memakan waktu sebanyak 254.00 menit. Pada data latih nilai *loss* yang diperoleh sebesar 0.037, nilai *dice* sebesar 0.963 dan nilai *jaccard* sebesar 0.946. Pada data validasi nilai *loss* yang diperoleh sebesar 0.037, nilai *dice* sebesar 0.962 dan nilai *jaccard* sebesar 0.928. Pada data uji nilai *dice* yang diperoleh sebesar 0.967 dan nilai *jaccard* sebesar 0.937.

Evaluasi yang dilakukan adalah membandingkan setiap keluaran dari setiap model seperti model itu sendiri, waktu pelatihan dan nilai *loss*. Model dibandingkan dengan cara prediksi segmentasi. Proses ini untuk melihat seberapa baik model dalam mengartikan masukan gambar dan keluarannya berupa masking dari gambar *x-ray*. Contoh perbandingan segmentasi setiap model pada salah satu gambar data uji bisa dilihat pada **Error! Not a valid bookmark self-reference.**

Model dengan waktu pelatihan tercepat adalah *U-net Mobilenet* dengan waktu selama 246.68 menit, kemudian *U-net InceptionV3* selama 250.25 menit, kemudian *U-net EfficientNetb0* selama 251.64 menit dan yang terakhir jurnal rujukan selama 254 menit. Proses faktorisasi pada layer *mobilenet* membuat proses komputasi semakin efisien dan mengurangi ukuran model. Perbandingan waktu pelatihan bisa dilihat pada Tabel 2.

Tabel 2. lama pelatihan setiap model

Model	Time (Minute)
<i>U-net Inception V3</i>	250.25
<i>U-net MobileNet</i>	246.68
<i>U-net EfficientNetb0</i>	251.64
Jurnal Rujukan	254.00

Pada data latih, Model dengan performa terbaik adalah *U-net Efficientnetb0* dengan nilai *dice* 0.972 dan *jaccard* 0.946. Perbandingan *dice* dan *jaccard* pada data latih setiap model bisa dilihat pada Tabel 3.

Tabel 3. Perbandingan Dice, Jaccard dan Loss data latih

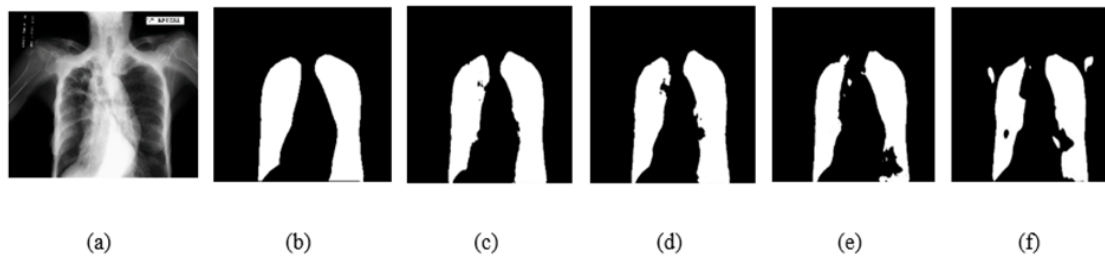
Model	Dice	Jaccard	Loss
<i>U-net Inception V3</i>	0.971	0.944	0.029
<i>U-net MobileNet</i>	0.970	0.943	0.03
<i>U-net EfficientNetb0</i>	0.972	0.946	0.028
Jurnal Rujukan	0.963	0.928	0.037

Pada data validasi, Model dengan performa terbaik adalah *U-net EfficientNetb0* dengan nilai *dice* 0.969 dan *jaccard* 0.940. Perbandingan *Dice* dan *Jaccard* pada data validasi setiap model bisa dilihat pada

Tabel 4.

Tabel 4. Perbandingan Dice, Jaccard dan Loss data validasi

Model	Dice	Jaccard	Loss
<i>U-net Inception V3</i>	0.968	0.938	0.032
<i>U-net MobileNet</i>	0.969	0.940	0.032
<i>U-net EfficientNetb0</i>	0.969	0.940	0.031
Jurnal Rujukan	0.962	0.928	0.037



Gambar 11. (a) gambar x-ray; (b) segmentasi manual; (c) *U-net InceptionV3*; (d) *U-net MobileNet*; (e) *U-net EfficientNetB0*; (f) jurnal rujukan

Pada data uji, model dengan performa terbaik adalah *U-net EfficientNetB0* dengan nilai *dice* sebesar 0.967 dan *jaccard* 0.937. Perbandingan *dice* dan *jaccard* pada data uji setiap model bisa dilihat pada Tabel 5.

Tabel 5. Perbandingan *dice* dan *jaccard* data uji

Model	<i>Dice</i>	<i>Jaccard</i>
<i>U-net InceptionV3</i>	0.964	0.931
<i>U-Net MobileNet</i>	0.964	0.932
<i>U-Net EfficientNetB0</i>	0.967	0.937
Jurnal Rujukan	0.963	0.93

4. Kesimpulan

Penelitian ini menggunakan metode *U-net* dikombinasikan dengan *pretrained model*, yaitu *InceptionV3*, *MobileNet*, dan *EfficientNetB0*. *U-net MobileNet* memiliki waktu tercepat dalam pelatihan model, disusul oleh *U-net InceptionV3*, *U-net efficientnetb0* dan yang terakhir model dari jurnal rujukan. Hasil segmentasi seluruh model pada gambar x-ray dapat dinilai bekerja dengan baik. Model dengan performa terbaik adalah *U-Net EfficientNetB0* dengan nilai *dice* 0.967 dan *jaccard* 0.937. Untuk penelitian selanjutnya penulis menganjurkan untuk menggunakan kombinasi *U-net* dengan *pretrained model* yang lain dan mengujinya menggunakan dataset yang sama atau berbeda.

Referensi

- [1] Zaitoun NM, Aqel MJ. Survey on Image Segmentation Techniques. *Procedia Comput Sci* [Internet]. 2015;65(Iccmit):797–806. Available from: <http://dx.doi.org/10.1016/j.procs.2015.09.027>
- [2] Minaee S, Boykov YY, Porikli F, Plaza AJ, Kehtarnavaz N, Terzopoulos D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans Pattern Anal Mach Intell*. 2021;8828(c):1–20.
- [3] Ghosh S, Das N, Das I, Maulik U. Understanding deep learning techniques for image segmentation. *ACM Comput Surv*. 2019;52(4).
- [4] Chowdhary CL, Acharjya DP. Segmentation and feature extraction in medical imaging: a systematic review. *Procedia Comput Sci*. 2020;167:26–36.
- [5] Rahmawati E, Wibawanto H. Sistem Pakar Diagnosis Penyakit Paru-Paru Menggunakan Metode Forward Chaining. *J Tek Elektro*. 2016;8(2):64–9.
- [6] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*. 2015. p. 234–41.
- [7] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p. 3431–40.
- [8] Islam J, Zhang Y. Towards robust lung segmentation in chest radiographs with deep learning. *arXiv Prepr arXiv181112638*. 2018;
- [9] Candemir S, Jaeger S, Palaniappan K, Musco JP, Singh RK, Xue Z, et al. Lung segmentation

- in chest radiographs using anatomical atlases with nonrigid registration. *IEEE Trans Med Imaging*. 2013;33(2):577–90.
- [10] Stirenko S, Kochura Y, Alienin O, Rokovyi O, Gordienko Y, Gang P, et al. Chest X-Ray Analysis of Tuberculosis by Deep Learning with Segmentation and Augmentation. 2018 IEEE 38th Int Conf Electron Nanotechnology, ELNANO 2018 - Proc. 2018;422–8.
- [11] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p. 1–9.
- [12] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv Prepr arXiv170404861*. 2017;
- [13] Tan M, Le Q V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv Prepr arXiv190511946*. 2019;
- [14] Yakubovskiy P. Segmentation Models. GitHub repository. GitHub; 2019.
- [15] Chaurasia A, Culurciello E. Linknet: Exploiting encoder representations for efficient semantic segmentation. In: *2017 IEEE Visual Communications and Image Processing (VCIP)*. 2017. p. 1–4.
- [16] Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p. 2881–90.
- [17] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. p. 4510–20.
- [18] Dice LR. Measures of the amount of ecologic association between species. *Ecology*. 1945;26(3):297–302.
- [19] Jaccard P. The distribution of the flora in the alpine zone. 1. *New Phytol*. 1912;11(2):37–50.

