

Performance Comparison of U-Net Input Preprocessing in Retina Vessel Segmentation Dataset

Karunia Perjuangan M
Departemen Teknik Elektro dan Teknologi Informasi
Universitas Gadjah Mada
Yogyakarta, Indonesia
karuniaperjuangan@gmail.com

Abstrak— Pembuluh darah retina merupakan bagian penting dari mata. Dengan perannya yang penting, berbagai diagnosis penyakit mata bisa dianalisis dari pola pembuluh darah retina. Proses pemisahan pola pembuluh darah retina dari citra yang dilakukan secara manual akan cukup merepotkan, terutama jika terdapat banyak citra yang harus dianalisis. Untuk itu, dirancanglah sistem segmentasi berbasis arsitektur U-Net yang mampu memudahkan proses pemisahan pola retina dari citra. Penelitian yang dilakukan bertujuan untuk meningkatkan performa dari model dengan mencari cara pengolahan masukan terbaik dari beberapa skenario yang diberikan. Performa dari model tersebut akan diukur dari metrik *IoU/Jaccard Index*, *F1-Score/Dice Index*, *Recall/Sensitivity*, *Precision/Specificity*, dan *Accuracy*.

Kata kunci—segmentasi, U-Net, image preprocessing, computer vision

I. LATAR BELAKANG

Pembuluh darah merupakan salah satu organ yang berperan penting dalam menyuplai aliran darah ke berbagai bagian tubuh, tidak terkecuali mata. Karena perannya yang penting itulah, berbagai diagnosis medis bisa diputuskan dari karakteristik pembuluh darah, termasuk tebal dan pola percabangan dari pembuluh darah itu sendiri [1]. Beberapa penyakit yang bisa didiagnosis dari karakteristik pembuluh darah antara lain hipertensi [1], glaukoma [2], dan diabetes [3]. Identifikasi pola retina menjadi bagian penting dari diagnosis penyakit tersebut.

Proses memisahkan pembuluh darah retina dari bagian lainnya dalam suatu citra disebut segmentasi. Segmentasi merupakan tahap penting dalam mengidentifikasi pola retina. Tahapan ini terkadang akan merepotkan bila terdapat banyak citra yang harus disegmentasi secara manual. Untuk itu, muncul beberapa penelitian yang mengenalkan berbagai cara melakukan segmentasi pembuluh darah pada retina.

Staal dkk. mengusulkan segmentasi berbasis *ridge* yang memanfaatkan pencarian *ridge* atau sebuah garis yang memiliki intensitas cahaya berbeda dari bagian sekitarnya [4]. Dari *ridge* ini, dibuatlah *feature vector* untuk masing-masing piksel dengan menghubungkan piksel tersebut dengan *ridge* terdekat. Setelah itu, dilakukan klasifikasi piksel menggunakan kNN untuk menentukan apakah piksel tersebut termasuk retina atau bukan.

Kemudian, muncul beberapa penelitian mengenai segmentasi pembuluh darah retina dengan memanfaatkan berbagai metode pengolahan citra, seperti *line tracking* [5], Adaboost [6], dan 2D Gabor Wavelet [7].

Pada tahun 2015, Ronnerberger mengusulkan sebuah arsitektur *convolutional network* untuk segmentasi yang disebut U-Net [8]. Arsitektur ini memanfaatkan dua bagian

jaringan yang disebut *contracting path* dan *expanding path*. Arsitektur ini menjadi populer dalam kasus segmentasi karena dapat bekerja dalam sedikit gambar dan performa yang dihasilkannya lebih tinggi dari penelitian-penelitian sebelumnya.

Pada penelitian ini, penulis mengusulkan adanya usaha peningkatan performa model berbasis U-Net dengan memodifikasi citra yang menjadi masukan model. Dengan mengubah masukan model, diharapkan terjadi peningkatan performa tanpa disertai peningkatan kompleksitas arsitektur dan jumlah parameter yang signifikan.

II. METODOLOGI PENELITIAN

A. Dataset

Dataset yang digunakan adalah *Digital Retinal Images for Vessel Extraction (DRIVE)*. Dataset ini berisi sampel citra retina yang telah dikelompokkan menjadi dua bagian, yaitu *training* dan *testing* yang masing-masing mengandung dua puluh sampel. Setiap sampel yang ada dalam dataset terdiri dari citra retina yang diambil menggunakan kamera Canon R5, pola retina yang dibuat secara manual (area retina ditandai dengan warna putih, sementara luar retina dinyatakan dengan warna hitam), serta *mask* yang menunjukkan bagian dari citra yang diambil menggunakan kamera [4]. Contoh sampel tersebut ditunjukkan oleh gambar berikut.

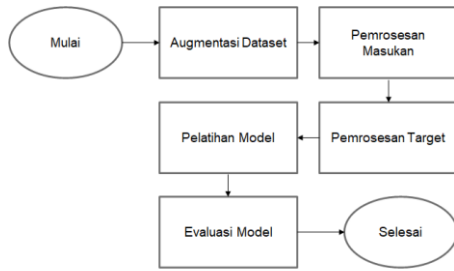


Gambar 1. Contoh sampel yang ada dalam dataset.

Citra yang ada dalam dataset memiliki dimensi 768x584 piksel. Dalam eksperimen kali ini, citra masukan diperkecil menjadi 128x128 piksel untuk memperingan proses pelatihan yang dilakukan dalam penelitian ini.

B. Metode Penelitian

Proses penelitian ini secara umum dapat diilustrasikan dalam diagram yang ada dalam gambar berikut:



Gambar 2. Alur kerja penelitian

Pada tahap awal penelitian, dilakukan proses augmentasi dataset untuk memperbanyak sampel yang menjadi bahan *training*. Hal ini dilakukan untuk mengurangi dampak *overfitting* yang terjadi karena sedikitnya data *training* yang hanya terdiri dari dua puluh sampel. Proses augmentasi dilakukan dengan melakukan *scaling*, *rotation*, dan *flipping* secara random sehingga terdapat sampel baru yang jumlahnya sepuluh kali lipat data *training* di awal.

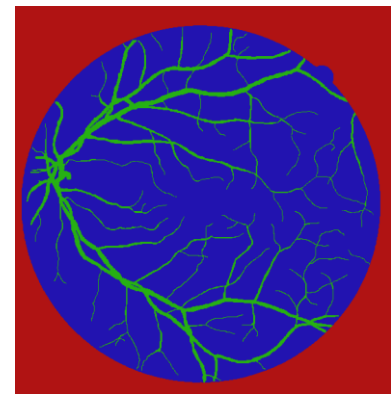
Kemudian dilakukan pemrosesan masukan dengan beberapa skenario. Skenario-skenario tersebut dibandingkan untuk mencari jenis pemrosesan terbaik.

No	Jenis Masukan	Jumlah Channel Masukan
1	Grayscale	1
2	Red Channel	1
3	Green Channel	1
4	Blue Channel	1
5	RGB	3
6	Canny Edge	1
7	Grayscale + Canny	2
8	RGB + Canny	4

Tabel 1. Beberapa skenario yang diuji beserta jumlah channel masukan.

Pemrosesan masukan ini dilakukan terhadap citra mata yang diambil dengan menggunakan kamera. Citra tersebut diproses dengan cara memilih atau memodifikasi *channel* tertentu sebagai masukan model. Selain itu, terdapat pula beberapa skenario yang memanfaatkan *channel* tambahan berupa hasil algoritma *Canny Edge Detection* yang mengambil masukan citra *grayscale* untuk menghasilkan citra baru yang merupakan hasil algoritma ini.

Selain itu, dilakukanlah modifikasi citra *ground truth* yang menggabungkan citra *mask* bola mata dan anotasi letak retina. Citra *ground truth* dapat diilustrasikan dalam gambar di bawah.



Gambar 3. Salah satu contoh citra *Ground Truth*

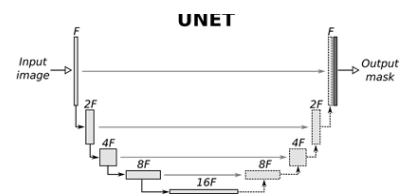
Dapat dilihat bahwa terdapat tiga jenis kelas yang ada dalam *ground truth*, yaitu latar belakang dalam mata yang ditunjukkan sebagai area berwarna biru, area retina mata yang ditunjukkan oleh warna hijau, dan juga latar belakang luar mata yang ditunjukkan dengan warna merah. Pemisahan latar belakang dalam mata dan luar mata bertujuan untuk mempermudah model dalam mengidentifikasi letak retina.

Setelah itu, dilakukanlah proses *training* model. Model yang digunakan sendiri berbasis model U-Net dengan *backbone* blok konvolusi biasa tanpa adanya koneksi residual dalam blok tersebut. Proses *training* dilakukan dengan meminimalkan nilai *Categorical Cross Entropy Loss* dengan menggunakan *optimizer* Adam.

Model yang telah selesai di-*train* akan dievaluasi menggunakan beberapa metrik, antara lain *Jaccard Index* (*Intersection over Union*), *F1-Score/Dice Index*, *Recall/Sensitivity*, dan *Precision/Specificity*.

C. Arsitektur

Arsitektur yang digunakan adalah U-Net yang dinamakan demikian karena bentuknya yang merupai huruf U. U-Net merupakan model yang sering digunakan dalam segmentasi citra biomedis. Salah satu kelebihan dari U-Net dibanding metode segmentasi tradisional adalah kemampuannya melakukan segmentasi dengan melihat fitur yang bersifat lokal dan juga fitur yang bersifat global secara bersamaan [8]. Kuncinya adalah menggunakan *skip connection* yang dapat mengurangi kontribusi *layer* yang dalam jika *layer* yang dangkal pun sudah dapat memberikan informasi yang cukup untuk proses segmentasi.



Gambar 4. Arsitektur U-Net

Arsitektur U-Net sendiri terdiri dari dua bagian, yaitu *contracting path* (sisi kiri huruf U) dan *expanding path* (sisi kanan huruf U). *Contracting Path* terdiri dari beberapa blok konvolusi yang secara perlahan-lahan mengecilkan dimensi citra yang sudah dikonvolusi dalam masing-masing bloknya dan melipatgandakan *channel*nya. Di akhir setiap blok dalam *contracting path*, dilakukanlah *pooling* untuk mengurangi dimensi. Jenis *pooling* yang digunakan dalam penelitian ini

adalah *max pooling*. Sementara itu, *Expanding Path* terdiri dari beberapa blok yang perlahan-lahan memperbesar dimensi citra dan memperkecil jumlah *channel*. Untuk memperbesar dimensi citra, digunakanlah *Upsampling Layer*.

Pada penelitian kali ini, jumlah *channel* pada blok terdangkal (setelah masukan) adalah 32, sementara jumlah *channel* pada blok terdalam adalah 512. Jumlah *channel* ini dipilih dengan pertimbangan bahwa dimensi citra awal berukuran 128x128 sehingga tidak diperlukan terlalu banyak *channel* yang dipakai untuk menganalisis segmentasi citra.

D. Metriks Evaluasi

Untuk menentukan skenario pemrosesan masukan terbaik dan memastikan bahwa model memiliki kemampuan segmentasi yang baik dari dataset, perlu dilakukan evaluasi dengan menggunakan beberapa metriks. Metriks ini diukur dari data validasi yang tidak ikut masuk sebagai data *training*. Hal ini perlu dilakukan untuk memastikan bahwa model bisa digunakan di data yang tidak masuk ke data *training*. Metriks tersebut terdiri dari:

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

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall/Sensitivity = \frac{TP}{TP + FN}$$

$$Precision/Specificity = \frac{TP}{TP + FP}$$

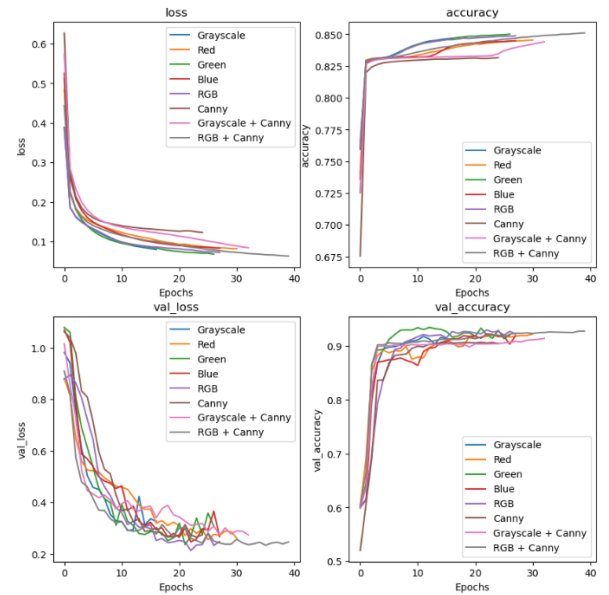
III. HASIL DAN ANALISIS

Bagian ini berfokus pada hasil evaluasi proses *training* beserta metriks yang dihitung dari data validasi. Untuk membandingkan performa model, ditentukanlah dua variabel bebas. Variabel bebas pertama adalah penggunaan jenis pemrosesan masukan seperti yang sudah ada dalam Tabel 1. Lalu, variabel bebas kedua adalah penggunaan augmentasi data dalam meningkatkan performa model dalam data yang ada di luar data *training*.

Proses *training* dilakukan sebanyak 40 *epoch* untuk dataset yang telah diaugmentasi sehingga terdapat sepuluh kali jumlah sampel dari dataset awal. Sementara itu, juga dilakukan *training* pada dataset yang tidak diaugmentasi sebanyak 400 *epoch* untuk mengkompensasi jumlah sampel yang lebih sedikit.

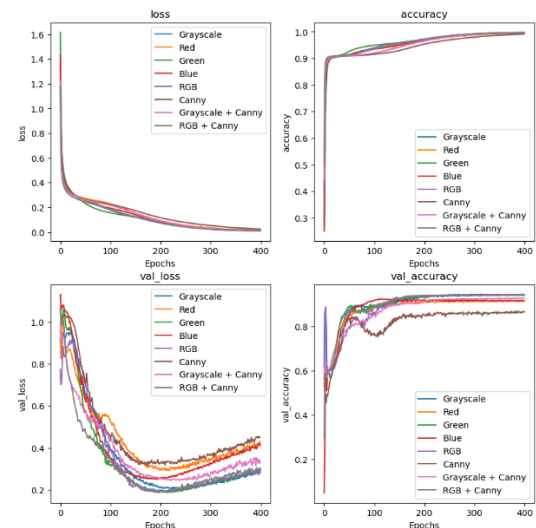
Training dilakukan dengan menggunakan *optimizer* Adam dengan *learning rate* $1e-4$ untuk meminimalkan *loss* *Categorical Cross Entropy* yang dipilih karena jumlah kelas pada *ground truth* lebih dari dua. Dilakukan *early stopping* dengan melihat perubahan *validation loss*. Jika *validation loss* stagnan atau meningkat dalam lima tahap, maka proses *training* dihentikan.

Proses *training* dan penurunan nilai *loss* dari dataset teraugmentasi dalam setiap *epoch*-nya terlihat dalam gambar berikut:



Gambar 5. Proses training pada dataset teraugmentasi

Sementara itu, proses *training* pada data tak teraugmentasi dapat diilustrasikan dalam gambar berikut:



Gambar 6. Proses training pada dataset tidak teraugmentasi

Dalam gambar di atas, terlihat bahwa jumlah *epoch* yang dibutuhkan model untuk mencapai titik optimum dalam dataset yang sama bernilai hampir sama. Namun, beberapa skenario memiliki titik *loss* terendah lebih rendah daripada beberapa skenario lain. Contohnya, skenario penggunaan “Green Channel” sebagai masukan memiliki *loss* lebih rendah daripada masukan “Canny”, “Red Channel”, dan “Blue Channel”. Artinya, informasi yang didapat dari “Green Channel” lebih membantu model melakukan segmentasi pada data validasi daripada *channel-channel* lainnya.

Hasil evaluasi metriks dari model berbasis dataset teraugmentasi ketika diuji ke dataset validasi dapat dilihat dalam tabel berikut:

	Jenis Masukan	Precision	Recall	F1_Score	Accuracy	IoU
1	Grayscale	0.979007	0.120215	0.214135	0.922699	0.119905
2	Red Channel	0.820421	0.270387	0.406728	0.930896	0.255279
3	Green Channel	0.978013	0.241718	0.387632	0.933093	0.240412
4	Blue Channel	0.950310	0.203191	0.334797	0.929263	0.201055
5	RGB	0.986939	0.205316	0.339918	0.930142	0.204759
6	Canny Edge	0.576613	0.004981	0.009877	0.912509	0.004963
7	Grayscale + Canny	0.946655	0.111889	0.200125	0.921643	0.111188
8	RGB+Canny	0.980661	0.254363	0.403950	0.934238	0.253093

Tabel 2. Beberapa skenario yang diuji beserta performanya pada dataset teraugmentasi.

Sementara itu, evaluasi metrik dari dataset yang tidak teraugmentasi dapat dilihat dalam tabel berikut:

	Jenis Masukan	Precision	Recall	F1_Score	Accuracy	IoU
1	Grayscale	0.767439	0.579475	0.660342	0.947775	0.492918
2	Red	0.536708	0.414080	0.467486	0.917355	0.305045
3	Green	0.720627	0.618107	0.665442	0.945551	0.498623
4	Blue	0.566410	0.423520	0.484653	0.921094	0.319830
5	RGB	0.741758	0.572125	0.645991	0.945065	0.477095
6	Canny	0.298386	0.338663	0.317251	0.872299	0.188531
7	Grayscale + Canny	0.707390	0.376772	0.491670	0.931747	0.325970
8	RGB + Canny	0.787955	0.523217	0.628860	0.945895	0.458640

Tabel 3. Beberapa skenario yang diuji beserta performanya pada dataset tidak teraugmentasi.

Dengan membandingkan kedua tabel, terlihat bahwa nilai *precision* dengan dataset teraugmentasi secara umum lebih besar daripada dataset yang tidak teraugmentasi. Namun hal yang menarik adalah semua nilai metrik selain *precision* justru turun setelah augmentasi dilakukan. Contohnya adalah nilai *recall* dari model yang awalnya bernilai sekitar 0,3 sampai 0,6 sebelum diaugmentasi justru turun menjadi sekitar 0-0,3 setelah diaugmentasi. Meningkatnya nilai *precision* dan menurunnya nilai *recall* ini dapat dilihat sebagai tanda lebih berhati-hatinya model dalam menganotasi retina. Karena itulah, banyak bagian retina yang seharusnya masuk ke pola segmentasi namun tidak terdeteksi oleh model. Sebaliknya, bagian yang bukan retina jarang sekali terdeteksi sebagai retina jika dilihat dari data *precision*. Salah satu hal yang mungkin menyebabkan penurunan performa model berbasis yang diaugmentasi adalah munculnya pola baru dari hasil *flipping* dan *rotation*. Pola ini tidak ditemukan dalam data validasi sehingga model tidak “berani” menyatakan suatu bagian sebagai retina.

Lalu, dengan melihat perbandingan antarskenario dalam tabel, kita dapat melihat bahwa penambahan *channel* Canny sebagai pelengkap *channel* lainnya justru mengurangi performa model. Misalnya, performa model dengan masukan

RGB + Canny lebih rendah dari model dengan masukan RGB saja. Artinya, Canny tidak memberikan kontribusi tambahan informasi yang membantu segmentasi. Dengan kata lain, informasi yang sebenarnya didapat dari Canny, yaitu letak edge secara tidak langsung sudah diketahui melalui proses-proses yang ada di dalam U-Net sehingga penggunaan *channel* Canny menjadi tidak diperlukan.

Terakhir, bila kita lihat dari beberapa skenario dalam tabel data *training* yang tidak teraugmentasi, performa paling baik dimiliki oleh skenario *Green Channel* yang paling unggul dari segi F1-Score, Recall, dan IoU. Skenario ini hanya kalah tipis dari skenario lainnya dalam bidang *precision* dan *accuracy*. Artinya, kontribusi informasi yang diberikan oleh *green channel* dalam RGB sendiri sudah cukup untuk membantu proses segmentasi. Penggunaan *green channel* bersama *channel-channel* lain justru menurunkan performa. Artinya, ketika kita ingin memaksimalkan performa model yang kita buat dalam *domain* ini, kita harus memfilter gambar sehingga hanya *channel Green* saja yang menjadi masukan model.

IV. KESIMPULAN

Model berbasis U-Net yang telah dibuat mampu menghasilkan citra segmentasi dengan mengolah citra masukan ke dalam berbagai blok yang ada dalam arsitektur U-Net. Performa yang dihasilkan oleh model sangat bergantung pada citra masukan. Citra masukan yang paling baik adalah citra yang hanya memasukkan “Green Channel” ke dalam masukan model. Penggunaan data augmentasi justru mengurangi performa model karena model menjadi lebih berhati-hati dalam menganotasi retina.

REFERENSI

- [1] M. K. Ikram, “Retinal vessel diameters and risk of hypertension: the Rotterdam Study, in Hypertension, vol 47, no. 2, pp 184-192, 2006
- [2] Chan, Karen KW, et al. "Retinal vasculature in glaucoma: a review." BMJ open ophthalmology 1.1 (2017): e000032.
- [3] T. Bek, “Diameter changes of retinal vessels in diabetic retinopathy” in Current diabetes reports, vol. 17, pp. 1-7, 2017
- [4] J. Staaf, M. D. Abramoff, M. Niemeijer, M. A. Viergever and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," in IEEE Transactions on Medical Imaging, vol. 23, no. 4, pp. 501-509, April 2004, doi: 10.1109/TMI.2004.825627.
- [5] M. Vlachos and E. Dermatas, “Multi-scale retinal vessel segmentation using line tracking” in Computerized Medical Imaging and Graphics, vol. 34, no. 3, pp. 213-227.
- [6] C. A. Lupascu, D. Tegolo and E. Trucco, "FABC: Retinal Vessel Segmentation Using AdaBoost," in IEEE Transactions on Information Technology in Biomedicine, vol. 14, no. 5, pp. 1267-1274, Sept. 2010, doi: 10.1109/TITB.2010.2052282.
- [7] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, H. F. Jelinek and M. J. Cree, "Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification," in IEEE Transactions on Medical Imaging, vol. 25, no. 9, pp. 1214-1222, Sept. 2006, doi: 10.1109/TMI.2006.879967.
- [8] O. Ronneberger, P. Fischer, & T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, pp. 234-241. Springer International Publishing, 2015