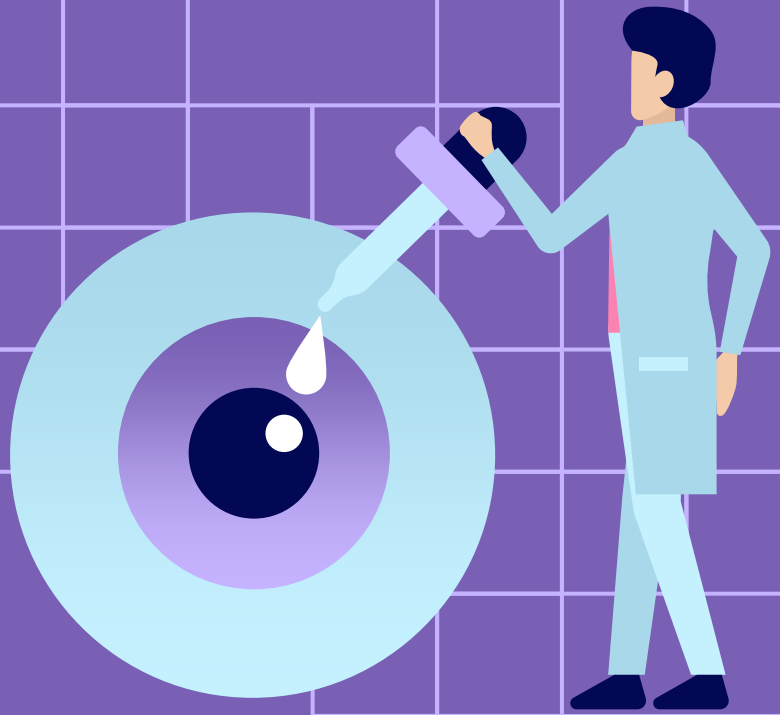
Performance Comparison of U-Net Input Preprocessing in Retina Vessel Segmentation Dataset

Karunia Perjuangan M. (20/456368/TK/50498)

TABLE OF CONTENTS

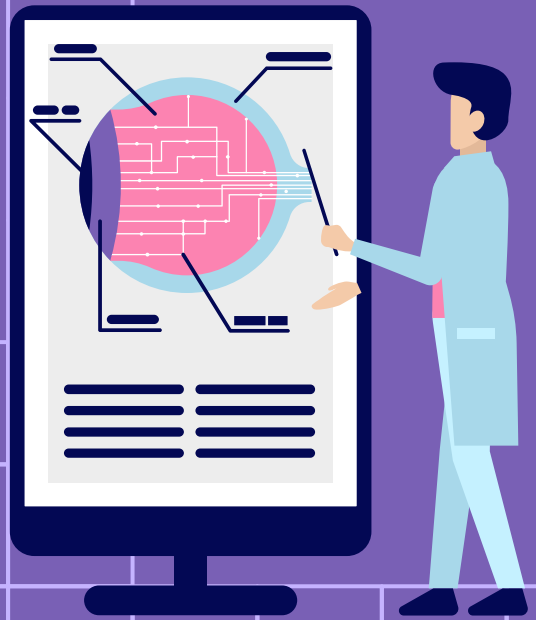
01

PENGANTAR



Latar Belakang Masalah

- Pembuluh darah adalah struktur yang berperan penting dalam jalannya suatu organ, tak terkecuali mata.
- Banyak penyakit yang mengubah pola pembuluh darah, seperti hipertensi, glaukoma, dan arteriosklerosis.
- Untuk membantu dokter mendiagnosis penyakit tersebut, **segmentasi citra** antara pembuluh darah dan objek lainnya perlu dilakukan agar dokter bisa melihat pola pembuluh darah dengan lebih jelas

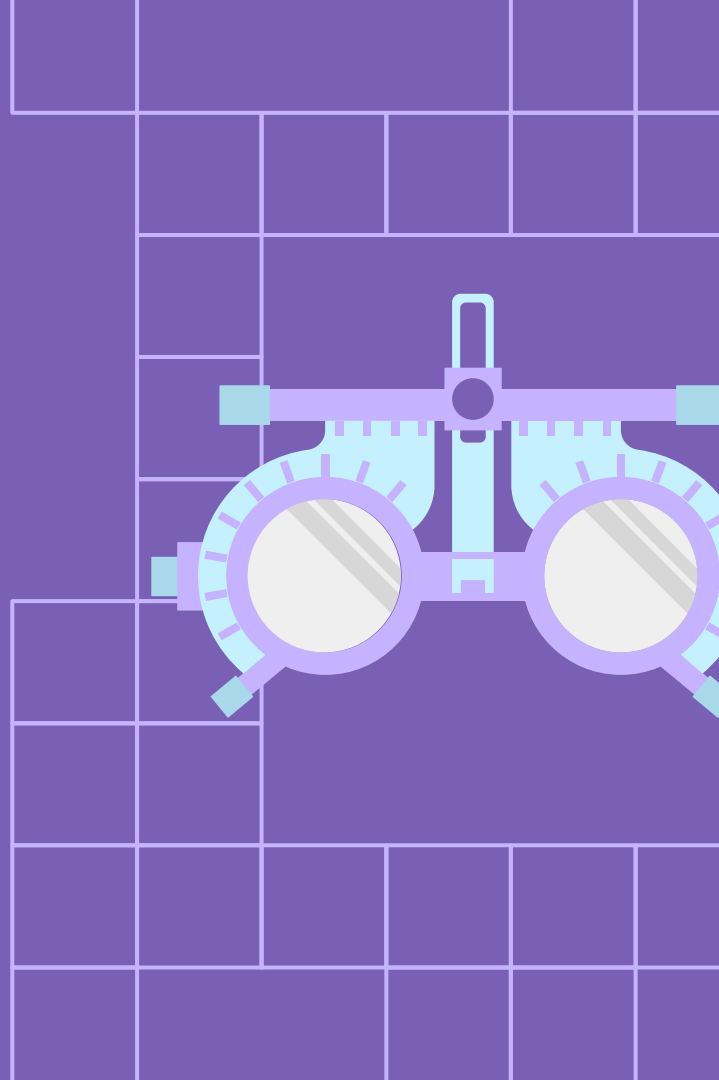


Latar Belakang Penelitian

- Salah satu arsitektur yang populer dalam segmentasi adalah **U-Net**
- U-Net mengambil input sebuah gambar dan berusaha melabeli masing-masing pikselnya dengan target mask.
- Untuk menghasilkan output terbaik, input apakah yang paling baik? Apakah grayscale saja (1 channel), RGB (3 channel), ataukah bisa dengan mengubah dulu dengan Filter, misalnya Sobel atau Canny? **Ataukah kombinasi dari Filtered dan Raw Image bisa membantu meningkatkan performa arsitektur U-Net?**

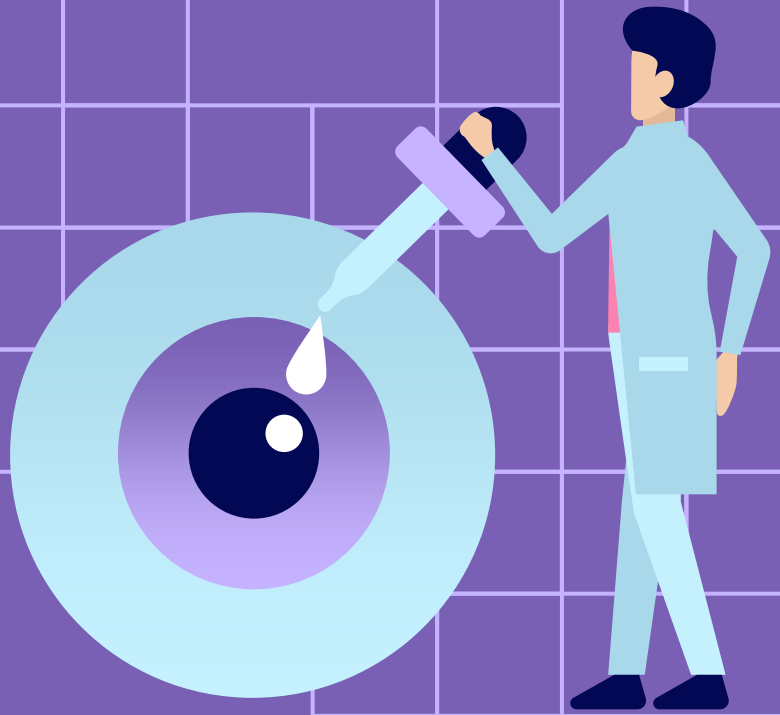
Tujuan Penelitian

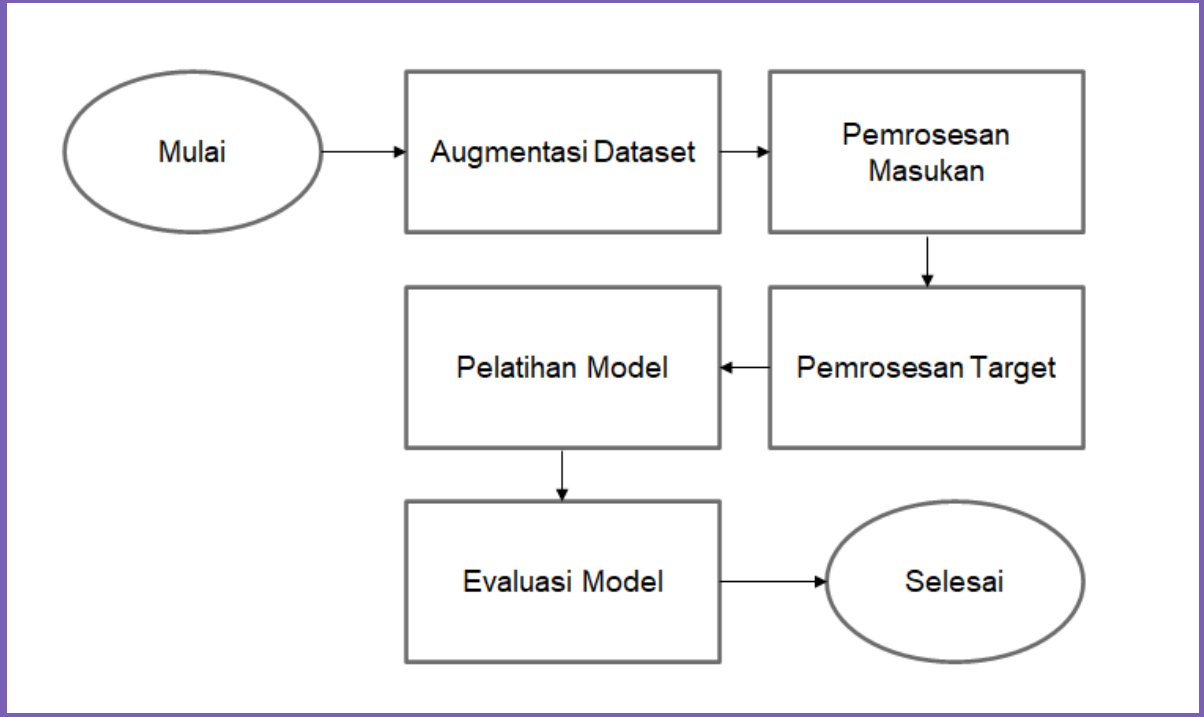
- Mencari input terbaik dari **U-Net**
- Melihat pengaruh berbagai jenis filter terhadap performa **U-Net**
- Meningkatkan performa U-Net agar tenaga medis lebih mudah mengetahui pola pembuluh darah dalam rangka diagnosis penyakit.



02

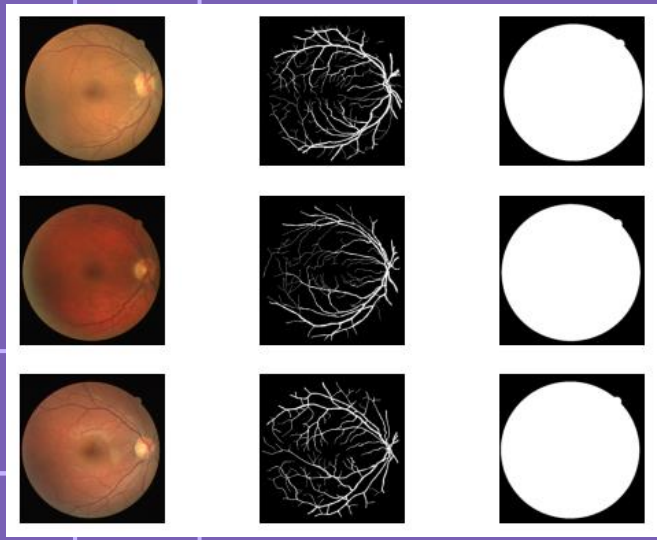
METODOLOGI




Mulai
Augmentasi Dataset
Pemrosesan Masukan
Pemrosesan Target
Pelatihan Model
Evaluasi Model
Selesai

Dataset

- 40 gambar fundus berwarna, 7 di antaranya masuk dalam kasus abnnormal.
- Dibagi dalam training set dan test set dalam jumlah yang sama (20-20)
- Ukuran piksel sebelum diresize adalah 584x565px dalam 3 channel RGB.
- Label berupa sebuah target mask yang dicirikan dengan warna putih untuk letak pembuluh darah dan warna hitam untuk latar belakang



Apa yang dibandingkan?

- Pemilihan Jenis Input

1. Grayscale (1 Channel)
2. Red Channel (1 Channel)
3. Green Channel (1 Channel)
4. Blue Channel (1 Channel)
5. RGB Channel (3 Channel)
6. Canny Operator (1 Channel)
7. Grayscale + Canny (2 Channel)
8. RGB + Canny (4 Channel)

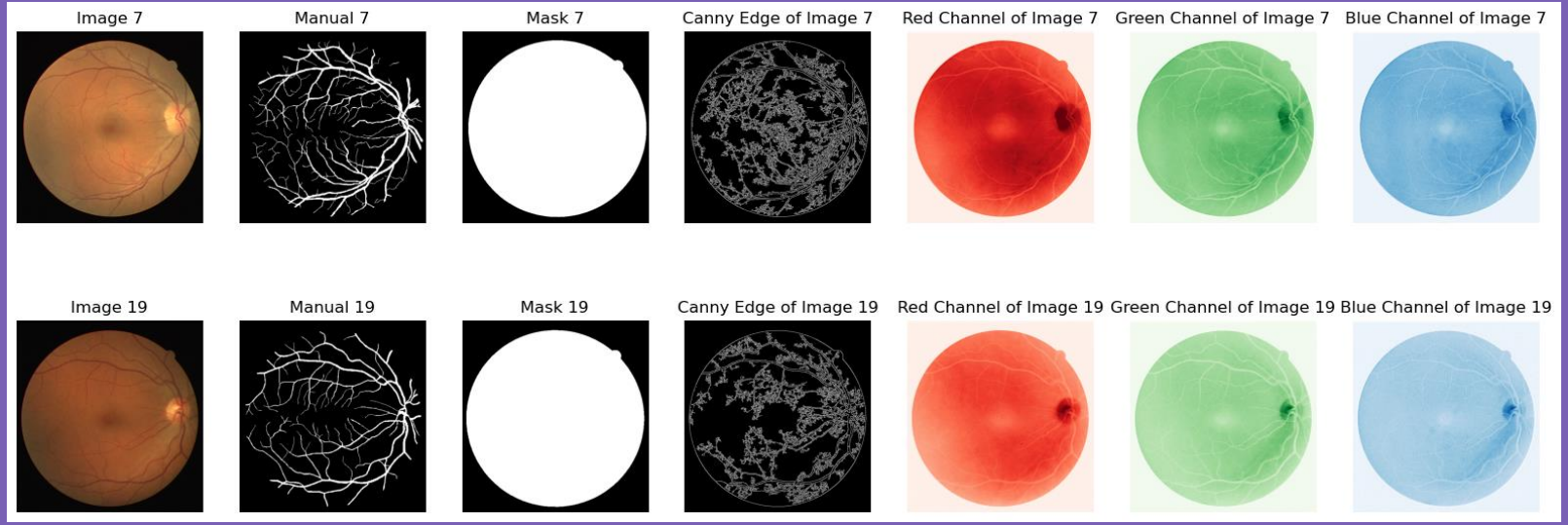
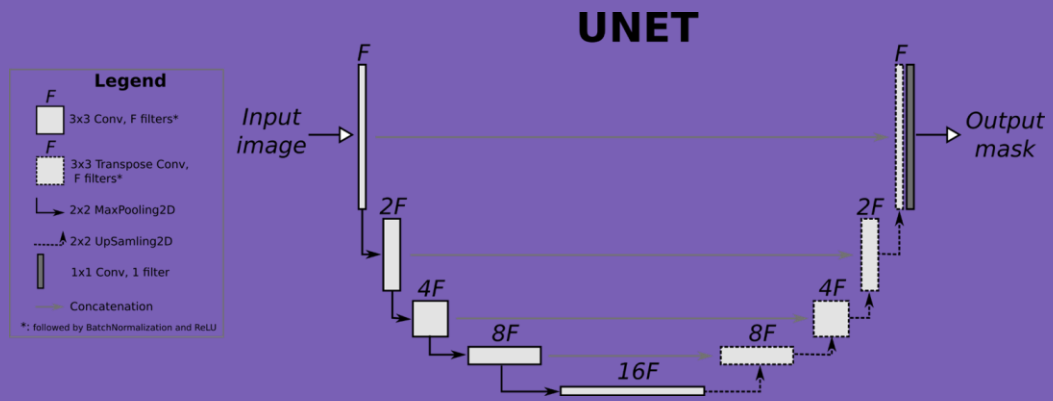


Image 7
Manual 7
Mask 7
Canny Edge of Image 7
Red Channel of Image 7
Green Channel of Image 7
Blue Channel of Image 7
Image 19
Manual 19
Mask 19
Canny Edge of Image 19
Red Channel of Image 19
Green Channel of Image 19
Blue Channel of Image 19

Setting & Hiperparameter

- Optimizer : Adam (lr=1e-4, beta1=0.9, beta2= 0.99)
- Loss : Sparse Categorical Cross Entropy (Ada 3 kelas: Pembuluh darah, Bukan Pembuluh Darah, Background)
- Epoch : 250
- Early Stop
- Batch Size : 1
- Arsitektur : Vanilla U-Net
- Resolusi : 128x128
- Threshold Canny : 0 & 75 (Handpicking)

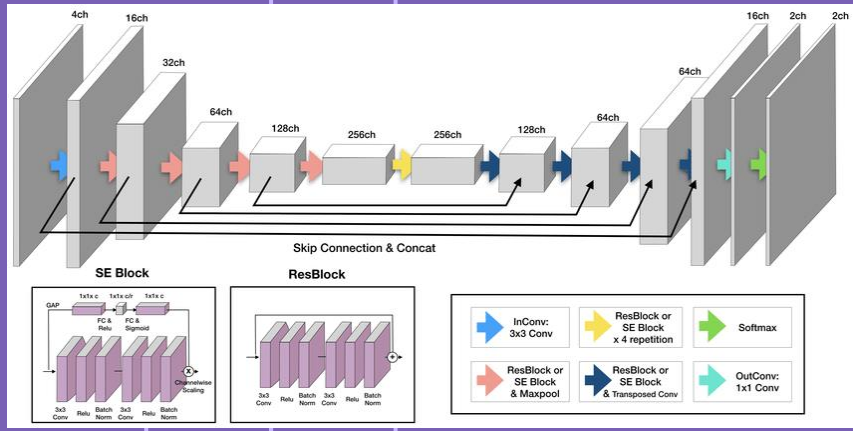
Arsitektur U-Net



- F diset 16
- Total parameter = ~8M (bervariasi sedikit karena perbedaan jumlah layer input per kasus)

Ciri Arsitektur U-Net

- Terdiri dari Expanding Layer dan Contracting Layer
- Di akhir expanding layer, dimensinya mengecil dengan Pooling dan channelnya dikali 2.
- Di awal contracting layer, terdapat proses Upsampling yang membuat dimensinya membesar. Hasil upsampling akan diconcatenate ke Expanding Layer dengan dimensi sama sebagai skip connection.

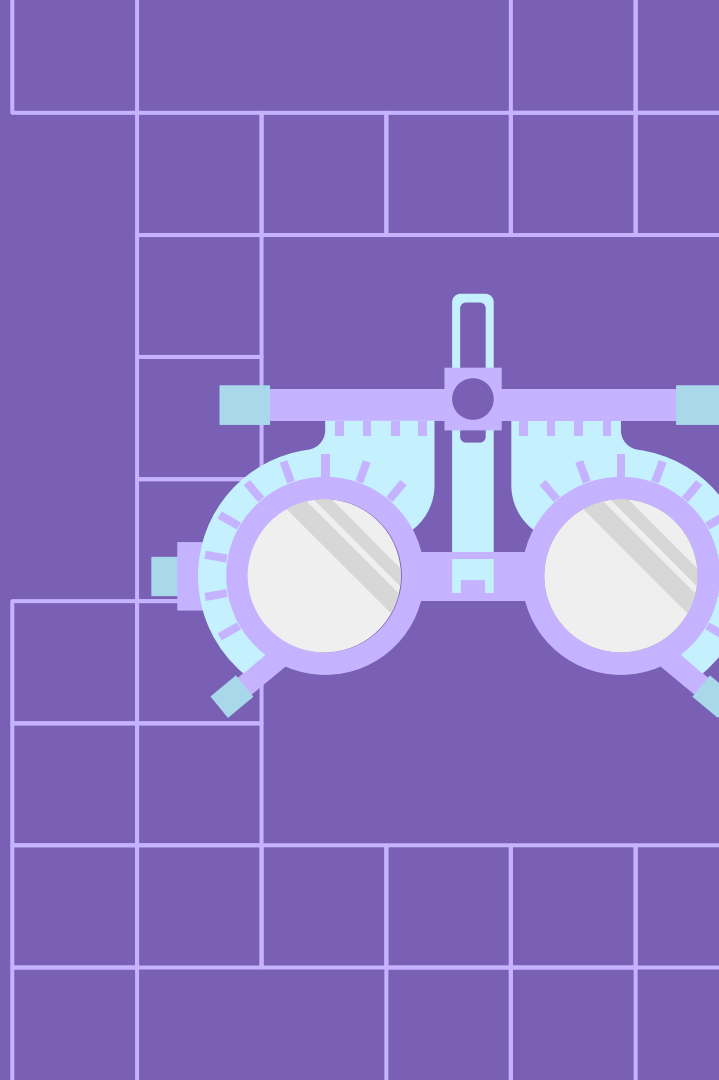


Metriks yang Digunakan

- Intersection over Union (IoU)
  Menggambarkan proporsi irisan antara target dan prediksi dibandingkan dengan gabungan keduanya
  $$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$
- Accuracy
  Menggambarkan proporsi nilai yang sama antara piksel prediksi dan piksel target.
  Acc = TP+TN/TP+TN+FP+FN

- Precision / Specificity
  Menggambarkan seberapa dapat dipercayanya hasil suatu positif suatu model
  Pre = TP/ TP+FP

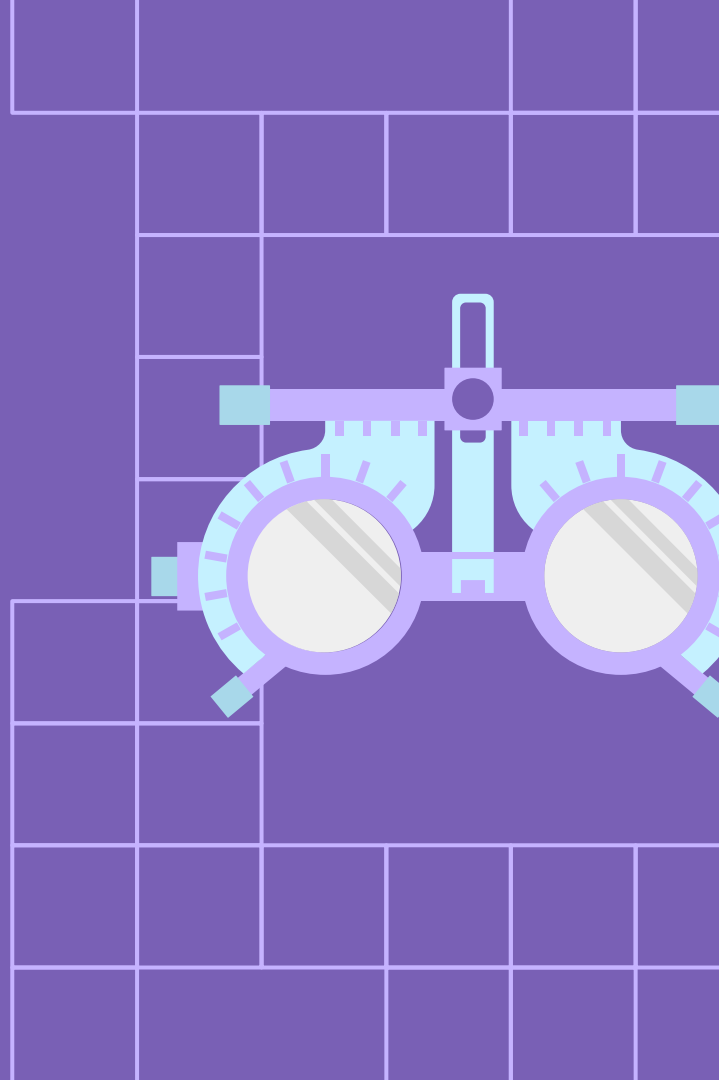
- Recall / Sensitivity
  Menggambarkan seberapa banyak label positif asli yang dapat dicari oleh model
  Rec = TP/TP+FN



Metriks yang Digunakan

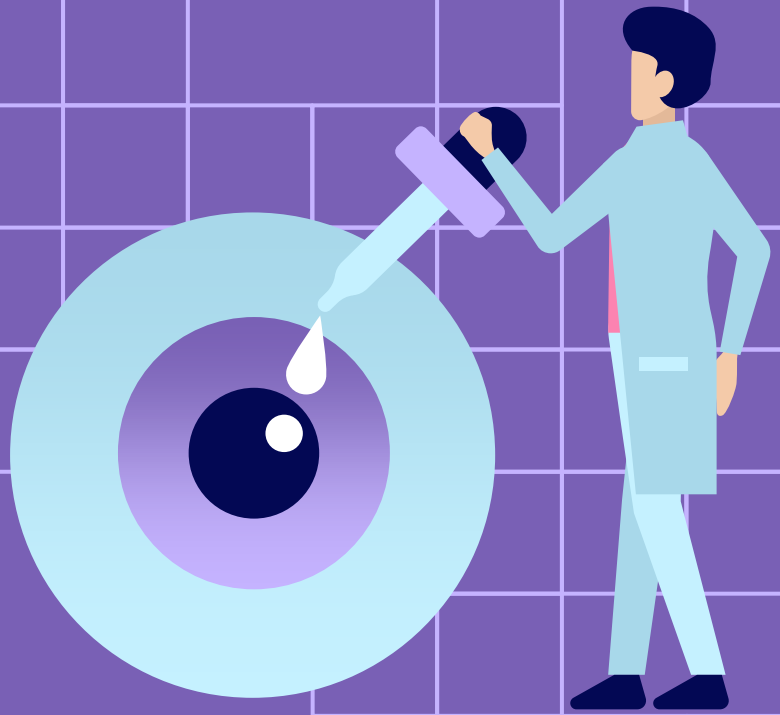
- F1-Score
  Menunjukkan akurasi pada kasus kelas tak berimbang

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{tp}}{2\mathrm{tp} + \mathrm{fp} + \mathrm{fn}}.$$

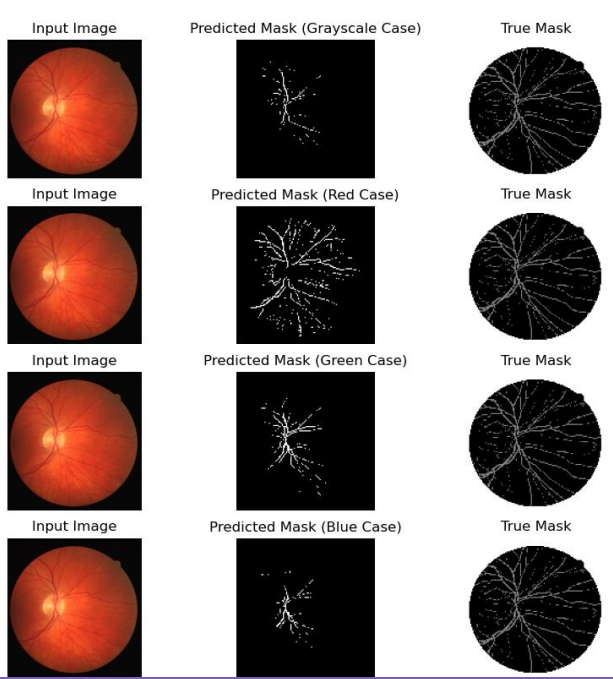


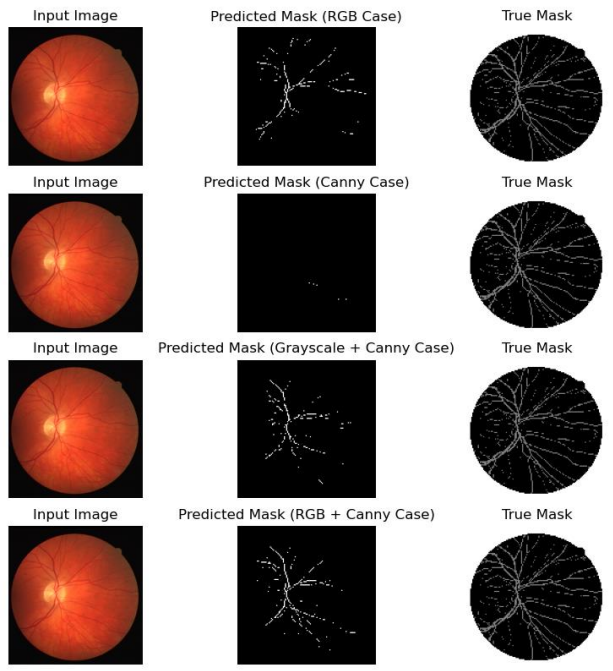
03

HASIL



Contoh Hasil Prediksi

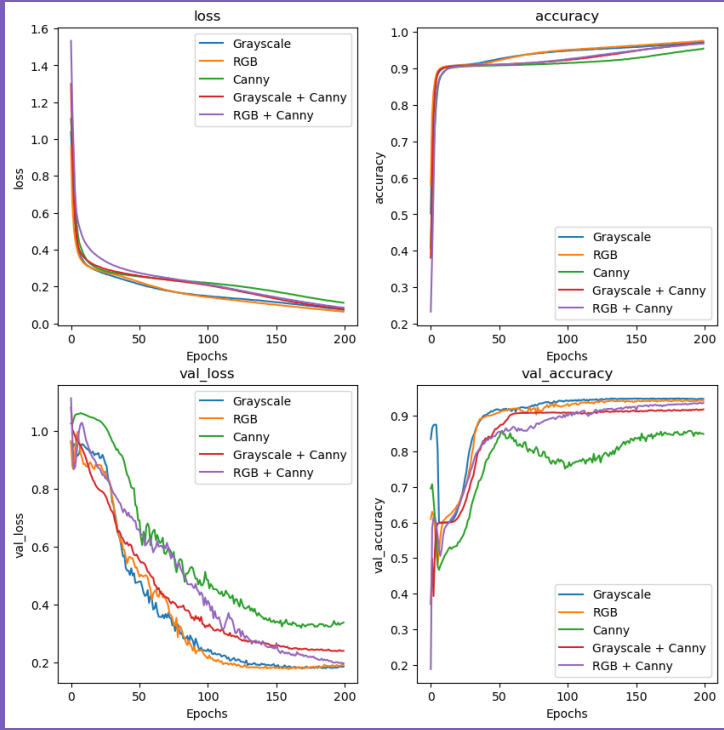




Hasil Metriks (Data Teraugmentasi)

	input	total_params	loss	val_loss	accuracy	val_accuracy
0	Grayscale	8635875	0.079597	0.326288	0.846819	0.911804
1	Red	8635875	0.081463	0.261090	0.845461	0.922351
2	Green	8635875	0.067785	0.308295	0.850109	0.920230
3	Blue	8635875	0.082760	0.267328	0.844864	0.918945
4	RGB	8636451	0.071945	0.248407	0.848842	0.922400
5	Canny	8635875	0.122711	0.273706	0.831626	0.904639
6	Grayscale + Canny	8636163	0.083934	0.273859	0.844037	0.913992
7	RGB + Canny	8636739	0.062909	0.246381	0.851026	0.927579

	input	precision	recall	f1_score	accuracy	jaccard
0	Grayscale	0.979007	0.120215	0.214135	0.922699	0.119905
1	Red	0.820421	0.270387	0.406728	0.930896	0.255279
2	Green	0.978013	0.241718	0.387632	0.933093	0.240412
3	Blue	0.950310	0.203191	0.334797	0.929263	0.201055
4	RGB	0.986939	0.205316	0.339918	0.930142	0.204759
5	Canny	0.576613	0.004981	0.009877	0.912509	0.004963
6	Grayscale + Canny	0.946655	0.111889	0.200125	0.921643	0.111188
7	RGB + Canny	0.980661	0.254363	0.403950	0.934238	0.253093


loss
Grayscale
RGB
Canny
Grayscale + Canny
RGB + Canny
loss
Epochs
accuracy
accuracy
Epochs
val_loss
val_loss
Epochs
val_accuracy
val_accuracy
Epochs

Hasil Metriks (Data Tidak diaugmentasi)

	input	total_params	loss	val_loss	accuracy	val_accuracy
0	Grayscale	8635875	0.009072	0.282867	0.996866	0.944348
1	Red	8635875	0.014171	0.409924	0.994928	0.914215
2	Green	8635875	0.009412	0.292133	0.996869	0.942075
3	Blue	8635875	0.012189	0.413873	0.995703	0.916956
4	RGB	8636451	0.010093	0.300128	0.996558	0.941772
5	Canny	8635875	0.024384	0.449219	0.990918	0.865903
6	Grayscale + Canny	8636163	0.010915	0.331775	0.996170	0.928616
7	RGB + Canny	8636739	0.009446	0.303589	0.996744	0.942831

	input	precision	recall	f1_score	accuracy	jaccard
0	Grayscale	0.767439	0.579475	0.660342	0.947775	0.492918
1	Red	0.536708	0.414080	0.467486	0.917355	0.305045
2	Green	0.720627	0.618107	0.665442	0.945551	0.498623
3	Blue	0.566410	0.423520	0.484653	0.921094	0.319830
4	RGB	0.741758	0.572125	0.645991	0.945065	0.477095
5	Canny	0.298386	0.338663	0.317251	0.872299	0.188531
6	Grayscale + Canny	0.707390	0.376772	0.491670	0.931747	0.325970
7	RGB + Canny	0.787955	0.523217	0.628860	0.945895	0.458640

Kesimpulan

- Penggunaan Filter Canny justru merendahkan kualitas model
- 1 Input Channel Grayscale sudah cukup untuk menjadi input segmentasi berbasis U-Net
- U-Net sudah secara otomatis mendeteksi adanya pola sehingga kita tidak perlu melakukan filter manual sebelum menjadi input

Kode

https://drive.google.com/file/d/1WeCM7uibXXplJnXNevU61wgvhEoF6m9O/view?usp=sharing