

KLASIFIKASI DIABETES RETINOPATI MENGGUNAKAN METODE *STATISTICAL REGION MERGING* DAN *CONVOLUTIONAL NEURAL NETWORK*

CLASSIFICATION OF DIABETIC RETINOPATHY USING STATISTICAL REGION MERGING AND CONVOLUTIONAL NEURAL NETWORK

Chintya Dewi Regina Wulandari¹, Suryo Adhi Wibowo, S.T., M.T., Ph.D.²,
Ledy Novamizanti, S.Si., M.T.³

^{1, 2, 3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
1chintyadewi@student.telkomuniversity.ac.id, 2suryoadhiwibowo@telkomuniversity.ac.id,
3ledyaldn@telkomuniversity.ac.id

Abstrak

Diabetes Retinopati merupakan salah satu gangguan penglihatan yang terjadi pada retina mata. Bila penyakit ini terlambat ditangani, penderita akan mengalami kebutaan. Peningkatan keparahan DR dapat dicegah jika dilakukan perawatan dan pemeriksaan dengan tepat. Pemeriksaan secara manual oleh dokter mata, akan menyita waktu dan hasil pemeriksaan bergantung pada keahlian dokter dalam mendiagnosis. Maka dari itu, diperlukan sebuah sistem yang dapat mendiagnosis secara cepat, akurat, dan objektif.

Berdasarkan permasalahan tersebut, pada Tugas Akhir ini penulis merancang suatu sistem yang dapat mengolah citra digital fundus mata dan mengklasifikasikannya berdasarkan tingkat keparahan DR. Tingkat keparahan DR pada Tugas Akhir ini dibagi menjadi 4 kelas yaitu normal, *mild* NPDR, *moderate* NPDR, dan *severe* NPDR. Data yang digunakan pada penelitian ini adalah data sekunder dari *dataset* MESSIDOR.

Sistem yang dirancang pada Tugas Akhir ini menggunakan metode segmentasi ciri *Statistical Region Merging* dan metode klasifikasi *Convolutional Neural Network*. Jumlah data yang digunakan adalah 80 citra, yang terdiri dari 20 sampel citra untuk masing-masing kelas. Nilai akurasi tertinggi yang diperoleh dari sistem ini adalah 81,25% dengan waktu komputasi 14,598 detik. Nilai akurasi tersebut didapatkan dengan menggunakan perbandingan 3:2 data latih dan data uji, nilai kompleksitas segmentasi $Q=256$, jumlah *epoch*=100 dan nilai *learning rate*=0,0001.

Kata Kunci: DR, NPDR, SRM, CNN.

Abstract

Diabetes Retinopathy is one of the complications of Diabetes Mellitus, this complication occurs in the retina of the eye. Patients will experience vision problems and if it handled late, patients will experience blindness. Manual examination by an ophthalmologist will take time and the results of the examination also depend on the doctor's expertise in diagnosing. Therefore, a digital image processing system which able diagnose quickly, accurately, and objectively, is needed.

Based on these problems, in this Final Project the author desing a system that can process digital fundus images and classify them into 4 classes namely normal, *mild* NPDR, *moderate* NPDR, and *severe* NPDR. The data used in this study are secondary data from the MESSIDOR dataset.

The data is processed using *Statistical Region Merging* (SRM) segmentation method and is classified using the *Convolutional Neural Network* (CNN) method. The number of data samples used in this thesis is 80 images, which consist of 20 image samples for each class. The highest accuracy value obtained from this system is 81.25% with a computational time of 14.518 seconds. The accuracy value is obtained by using a ratio of 3:2 training data and test data, the value of segmentation complexity parameters $Q=256$, the number of *epochs*=100 and the *learning rate*=0.0001.

Keywords: DR, NPDR, SRM, CNN.

1. Pendahuluan

Diabetes Retinopati (DR) merupakan salah satu komplikasi penyakit Diabetes Melitus. Komplikasi ini terjadi pada bagian retina mata. Penderita akan mengalami gangguan penglihatan dan apabila terlambat ditangani, penderita akan mengalami kebutaan. *International Diabetes Management Practices Study*, menyebutkan sebanyak 34,6% penderita DR merupakan populasi dari penderita Diabetes Melitus. Diestimasi 8,8 juta orang yang terkena diabetes di Indonesia, diprediksi akan meningkat menjadi 21,3 juta di tahun 2030.

Seiring dengan meningkatnya angka penderita DR, maka perlu diimbangi dengan peningkatan kepedulian masyarakat akan gejala dan komplikasi yang akan diakibatkan oleh penyakit ini. Pada penderita DR, pemeriksaan fundus yang teratur sangat penting, diikuti dengan perawatan laser yang sesuai jika diperlukan. Hal ini bertujuan untuk mencegah komplikasi semakin parah [1]. Sebuah studi mengatakan bahwa 90% dari kasus meningkatnya keparahan DR dapat dicegah jika dilakukan perawatan dan pemeriksaan mata pasien dengan tepat [2]. Pemeriksaan secara manual oleh dokter mata, tentunya akan menyita waktu dan hasil pemeriksaan bergantung pada keahlian dokter dalam mendiagnosis. Oleh sebab itu, diperlukan sebuah sistem yang dapat mendiagnosis secara cepat, akurat, dan objektif.

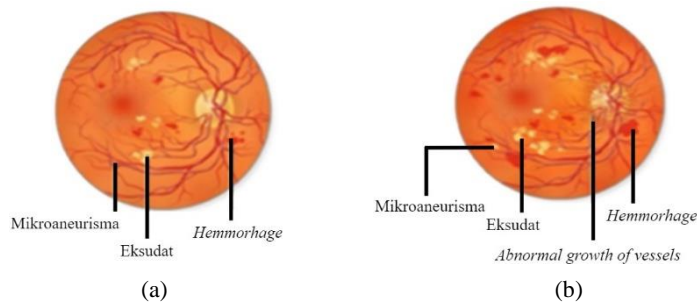
Berdasarkan permasalahan-permasalahan yang disebutkan di atas, pada Tugas Akhir ini penulis mengembangkan penelitian-penelitian sebelumnya dengan mengklasifikasikan empat tingkat keparahan DR, yaitu normal, *mild* NPDR, *moderate* NPDR dan *severe* NPDR dengan metode segmentasi *Statistical Region Merging* (SRM) dan metode klasifikasi *Convolutional Neural Networks* (CNN) yang menggunakan *transfer learning*

2. Dasar Teori

2.1 Diabetes Retinopati

DR merupakan salah satu komplikasi penyakit diabetes. Komplikasi tersebut berupa kerusakan pada bagian retina mata yang akan berdampak langsung pada terganggunya penglihatan penderita dan apabila terlambat ditangani akan menyebabkan penderita mengalami kebutaan permanen [3]. Secara umum tingkat DR dibedakan menjadi 2 jenis, yaitu *Nonproliferative Diabetic Retinopathy* (NPDR) dan *Proliferative Diabetic Retinopathy* (PDR).

NPDR adalah *stage* awal dari DR, yang dibagi menjadi tiga tingkat keparahan yaitu *mild* NPDR, *moderate* NPDR dan *severe* NPDR. Sedangkan PDR adalah *stage* yang lebih parah daripada NPDR. Gambar 1. adalah ilustrasi perbandingan NPDR dan PDR.



Gambar 1. Perbandingan (a) NPDR dan (b) PDR

2.2 Perancangan Sistem dan Realisasi Sistem

Pada Tugas Akhir ini dirancang sistem yang dapat mengklasifikasikan citra fundus mata berdasarkan tingkat keparahan DR, menjadi 4 kelas yaitu normal, *mild* NPDR, *moderate* NPDR dan *severe* NPDR. Data yang digunakan pada Tugas Akhir ini diambil dari dataset MESSIDOR, berjumlah 80 citra fundus yang terdiri dari 20 citra fundus untuk masing-masing kelas. Seluruh data akan melalui tahap *preprocessing* yaitu *cropping*, *resize* dan segmentasi ciri. Metode segmentasi ciri yang digunakan pada Tugas Akhir ini adalah SRM, sedangkan metode yang digunakan untuk klasifikasi adalah CNN. Gambaran sistem yang akan dirancang pada Tugas Akhir ini dapat dilihat pada Gambar 2.



Gambar 1. Diagram Alir Preprocessing

2.2 MESSIDOR DATASET

Dataset MESSIDOR telah dipublikasikan sejak 2008. Proyek ini didanai oleh program *Techno-Vision* Prancis. Dataset ini berisi 1200 citra mata fundus yang didapatkan dari 3 departemen *ophthalmologic* menggunakan *color video 3CCD camera*. Masing-masing citra berukuran 1440×960, 2240×1448, atau 2304×1536 piksel. Citra ini dikelompokkan berdasarkan dari departemen *ophthalmological* mana citra tersebut didapatkan. Setiap set, dibagi menjadi 4 zip sub set yang berisi 100 citra berformat .TIFF dan file Excel yang berisi diagnosa medis untuk setiap citra [4].

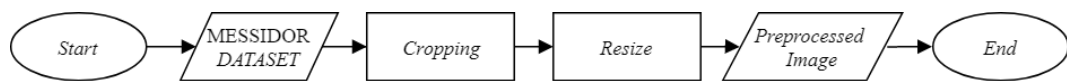
Data yang digunakan pada Tugas Akhir berjumlah 80 sampel data yang diambil dari *dataset* MESSIDOR. Sejumlah data tersebut terdiri dari citra fundus mata yang telah terklasifikasi normal, *mild NPDR*, *moderate NPDR* dan *severe NPDR*. Selanjutnya sampel data ini dibagi menjadi dua kelompok yaitu, data latih dan data uji. Sistematika jumlah sampel data yang digunakan dapat dilihat pada Tabel 1.

Table 1. Sistematika Jumlah Sampel Data

| Class | Amount of Data |
|-----------------------|----------------|
| <i>Normal</i> | 20 |
| <i>Mild NPDR</i> | 20 |
| <i>Moderate NPDR</i> | 20 |
| <i>Severe NPDR</i> | 20 |
| Amount of Data | 80 |

2.3 Preprocessing

Preprocessing adalah langkah awal yang diperlukan dalam pengolahan citra digital. *Preprocessing* dilakukan untuk meningkatkan kualitas citra sehingga citra yang diolah mendapatkan hasil yang maksimal saat melalui tahap ekstraksi ciri dan klasifikasi. Seluruh citra yang digunakan pada Tugas Akhir ini akan melalui tahap *preprocessing*, yaitu *cropping* dan *resize*. Seperti yang ditunjukkan oleh Gambar 3.



Gambar 3. Diagram Alir *Preprocessing*

2.4 Statistical Region Merging

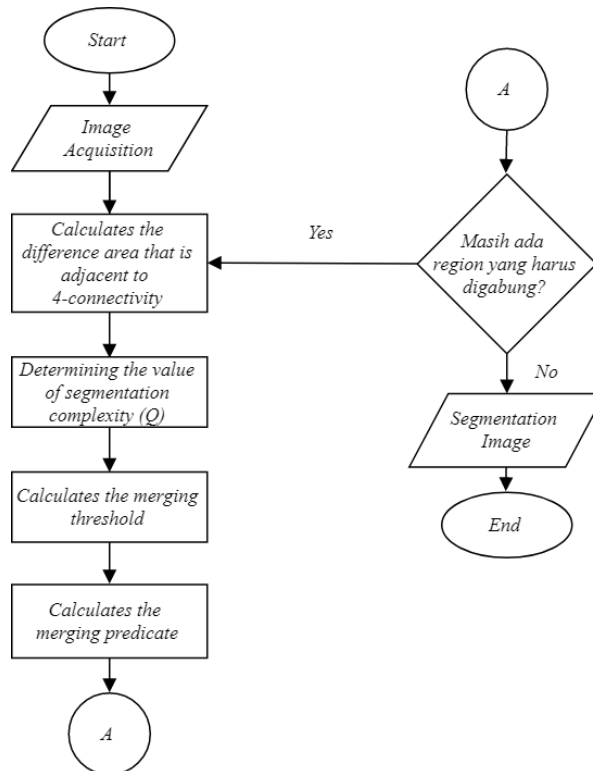
Statistical Region Merging (SRM) merupakan proses segmentasi citra yang membagi citra ke dalam beberapa daerah/region dengan setiap objek atau daerah memiliki kemiripan atribut. Metode segmentasi ini dikembangkan untuk segmentasi warna karena lebih cepat dan dapat mengurangi biaya komputasi. Metode ini sejenis dengan teknik *region growing and merging*. Pada *region growing*, region adalah kumpulan piksel yang homogen dan secara iterasi merupakan hasil gabungan dari region yang lebih kecil. Algoritma SRM ini didasarkan pada analisis statistik dari piksel yang berdekatan, lalu dilakukan penggabungan piksel yang memiliki kemiripan dalam suatu batasan atau region. Penggabungan piksel dilakukan jika memenuhi syarat [5]:

$$P(R, R') = \begin{cases} \text{true} & \text{if } \forall a \in \{R, G, B\}, \\ & |(R_a - R'_a)| \leq \sqrt{b^2(R) + b^2(R')} \\ \text{false} & \text{otherwise} \end{cases} \quad (1)$$

dimana $P(R, R')$ adalah nilai *merging predicate* dan \bar{R}_a, \bar{R}'_a adalah nilai rata-rata *channel a* di seluruh region R. Pada persamaan (1) terdapat $b(R)$ atau *merging threshold*, dimana nilai tersebut dapat dihitung dengan persamaan,

$$b(R) = g \sqrt{\frac{1}{2Q|R|} \ln \frac{2}{\delta}} \quad (2)$$

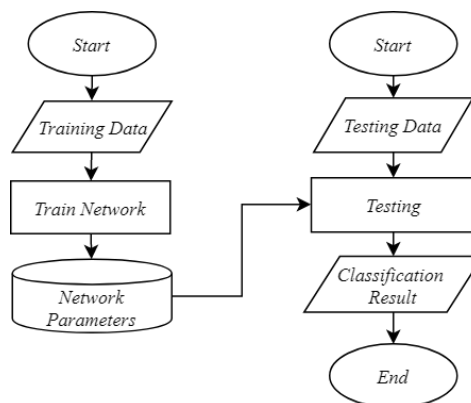
dimana g adalah nilai intensitas citra dan Q adalah parameter kompleksitas segmentasi. Nilai $\delta = \frac{1}{6|I|^2}$, dimana I adalah nilai piksel. Gambar 4 adalah diagram alir segmentasi ciri SRM.



Gambar 4. Diagram Alir Segmentasi Ciri SRM

2.5 Klasifikasi

Klasifikasi adalah tahap terakhir pada desain sitem ini. DR akan diklasifikasikan berdasarkan empat tingkat keparahan, yaitu normal, *mild NPDR*, *moderate NPDR* dan *severe NPDR*. Metode klasifikasi yang digunakan pada Tugas Akhir ini adalah *Convolutional Neural Network (CNN)* dengan *transfer learning AlexNet’s pretrained network*. CNN adalah pengembangan dari *Multi Layer Preceptron (MLP)*. Diagram alir klasifikasi ditunjukkan oleh Gambar 5.

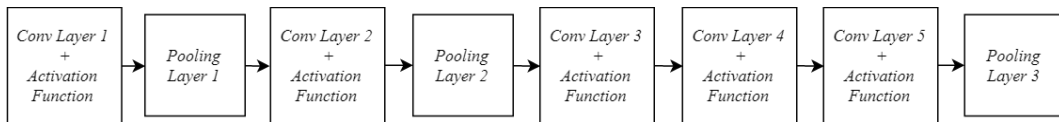


Gambar 5. Diagram Alir Klasifikasi

Secara umum, CNN terbagi menjadi dua bagian, yaitu *layer* ekstraksi ciri dan *layer* klasifikasi.

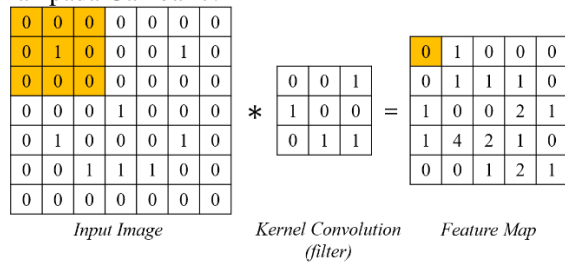
1) *Layer* Ekstraksi Ciri

Layer ekstraksi ciri, letaknya berada pada awal arsitektur. *Layer* ini tersusun atas beberapa *layer* dan setiap *layer* tersusun atas neuron yang terkoneksi pada daerah lokal (*local region*) *layer* sebelumnya. *Layer* jenis pertama adalah *layer* konvolusi dan *layer* kedua adalah *layer* pooling. Fungsi aktivasi ReLU diterapkan pada setiap *layer* konvolusi pada CNN [6]. Urutan *layer* ekstraksi ciri pada CNN diilustrasikan dengan Gambar 6.



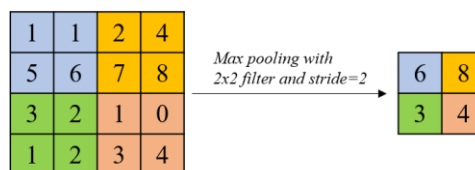
Gambar 6. Layer Ekstraksi Ciri

- *Convolution Layer* adalah bagian inti dari CNN. Konvolusi citra dilakukan untuk mendapatkan/mengekstraksi fitur dari citra *input* [7]. Ilustrasi *layer* konvolusi direpresentasikan pada Gambar 7.



Gambar 7. Operasi Konvolusi

- *Pooling Layer* berfungsi untuk mereduksi ukuran citra dan jumlah parameter dalam jaringan, untuk mempercepat komputasi dan mencegah terjadinya *overfitting*. *Pooling layer* yang umumnya digunakan pada CNN adalah *max pooling*. *Max pooling* akan mengambil nilai maksimum dari setiap matriks. Seperti yang direpresentasikan pada Gambar 8.

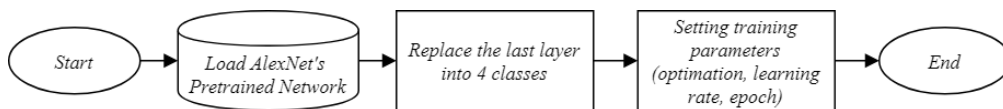


Gambar 8. Operasi Max Pooling

2) *Layer* Klasifikasi

Layer klasifikasi tersusun atas beberapa *layer* dengan banyak neuron yang terhubung secara menyeluruh (*fully connected*) dengan *layer* lainnya. *Layer* ini menerima *input* berupa vektor dari hasil *output layer* ekstraksi ciri. Hasil *output* dari *layer* ini berupa skoring kelas untuk klasifikasi [8].

Transfer learning digunakan pada pelatihan CNN, bermaksud untuk mengatasi keterbatasan *requirement* dan waktu komputasi yang tinggi. *Transfer learning* adalah proses penggunaan kembali sebuah *pretrained network* untuk menyelesaikan suatu masalah yang baru, dengan cara memodifikasi parameter sehingga sesuai dengan kebutuhan *dataset* yang baru [9]. *Pretrained network* yang digunakan pada Tugas Akhir ini adalah *pretrained network* AlexNet. Gambar 9 adalah diagram alir *transfer learning*.



Gambar 9. Diagram Transfer Learning

3. Pembahasan

Pada Tugas Akhir ini, performansi sistem diukur berdasarkan parameter akurasi dan waktu komputasi. Pengujian dilakukan menggunakan citra yang telah melalui tahap *preprocessing*. Total citra fundus yang digunakan berjumlah 80 sampel dengan rincian 20 sampel untuk masing-masing kelas. Pengujian sistem ini terdiri dari lima tahap yaitu:

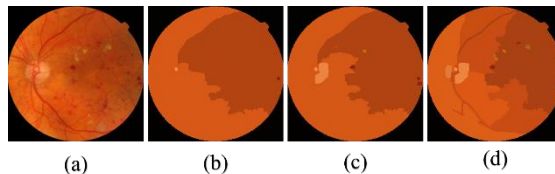
3.1 Pengujian Pengaruh Kompleksitas Segmentasi SRM (Q)

Pengujian ini dilakukan untuk mengetahui pengaruh nilai kompleksitas segmentasi (Q) terhadap nilai akurasi dan waktu komputasi. Pengujian pada tahap ini menggunakan nilai Q=64, 128 dan 256, jumlah *epoch*=20 dan nilai *learning rate*=0,001. Hasil pengujian ditunjukkan pada Tabel 2.

Tabel 2. Pengaruh Kompleksitas Segmentasi SRM (Q) terhadap Akurasi dan Waktu Komputasi

| Q | Training Data | | Testing Data | |
|-----|---------------|------------------------------|--------------|------------------------------|
| | Accuracy (%) | Computational Time (seconds) | Accuracy (%) | Computational Time (seconds) |
| 64 | 100 | 16.698 | 57.5 | 15.857 |
| 128 | 100 | 17.237 | 57.5 | 16.882 |
| 256 | 100 | 20.840 | 65 | 17.011 |

Berdasarkan Tabel 2. akurasi tertinggi dalam pengujian nilai Q adalah 65% dengan waktu komputasi 17,011 detik menggunakan nilai Q=256. Hal ini terjadi karena, semakin besar nilai Q, semakin kompleks juga representasi citra segmentasi yg dihasilkan. Sehingga sistem juga semakin banyak mempelajari ciri dan nilai akurasi yang diperoleh pun akan semakin tinggi. Hal ini dapat dilihat pada Gambar 10. yang menunjukkan perbandingan citra dengan nilai Q yang berbeda.



Gambar 10. Perbandingan Citra Hasil Segmentasi Ciri dengan Nilai Q yang Berbeda
(a) Citra Asli; (b) Citra Q=64; (c) Citra Q=128 dan (d) Citra Q=256

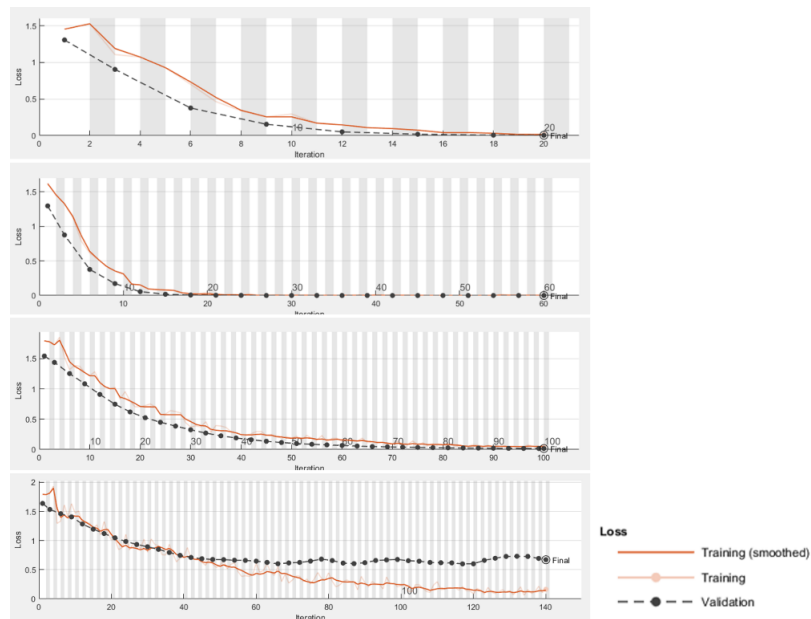
3.2 Pengujian Pengaruh Jumlah Epoch

Pengujian ini dilakukan untuk mengetahui pengaruh jumlah *epoch* terhadap akurasi dan waktu komputasi. Pengujian ini menggunakan jumlah *epoch*=20, 60, 100 dan 140, nilai Q=256, dan nilai *learning rate*=0,001. Hasil pengujian pengaruh *epoch* ditunjukkan oleh Tabel 3.

Tabel 3. Pengaruh Jumlah Epoch terhadap Akurasi dan Waktu Komputasi

| Epochs | Training Data | | Testing Data | |
|--------|---------------|------------------------------|--------------|------------------------------|
| | Accuracy (%) | Computational Time (seconds) | Accuracy (%) | Computational Time (seconds) |
| 20 | 100 | 20.840 | 65 | 17.011 |
| 60 | 100 | 18.625 | 65 | 17.678 |
| 100 | 100 | 16.250 | 70 | 17.621 |
| 140 | 100 | 21.775 | 67.5 | 19.068 |

Seperti yang terlihat pada Tabel 3., hasil pengujian pengaruh jumlah *epoch* menunjukkan nilai akurasi yang fluktuatif. Akurasi tertinggi dalam pengujian jumlah *epoch* adalah 70% dengan waktu komputasi 17,621 detik, menggunakan jumlah *epoch*=100. Pada penelitian ini, nilai *epoch*=100 sudah mencapai titik stabil. Hal ini dikarenakan ketika jumlah epoch terlalu sedikit, hal ini akan menyebabkan jaringan mengalami *underfitting*, seperti yang dapat dilihat saat nilai *epoch*=20 dan 60 pada Gambar 10. Sedangkan, ketika nilai *epoch* terlalu banyak, hal ini akan menyebabkan *overfitting*, seperti yang dapat dilihat pada nilai *epoch*=140 pada Gambar 11. Maka, dapat dikatakan bahwa jumlah *epoch* yang banyak, tidak menjadikan sebuah jaringan CNN akan lebih baik dalam proses pelatihan.



Gambar 11. Pengaruh Jumlah Epoch

3.3 Pengujian Pengaruh Nilai Learning Rate

Pengujian ini dilakukan untuk mengetahui pengaruh nilai *learning rate* terhadap akurasi dan waktu komputasi. Nilai *learning rate* yang diuji dalam penelitian ini adalah 0,001 dan 0,0001. Pengujian ini menggunakan nilai $Q=256$ dan nilai *epoch* 100. Tabel 4. menunjukkan hasil pengujian nilai *learning rate* terhadap akurasi dan waktu komputasi.

Tabel 4. Pengaruh *Learning Rate* terhadap Akurasi dan Waktu Komputasi

| <i>Learning Rate</i> | <i>Training Data</i> | | <i>Testing Data</i> | |
|----------------------|----------------------|-------------------------------------|---------------------|-------------------------------------|
| | <i>Accuracy (%)</i> | <i>Computational Time (Seconds)</i> | <i>Accuracy (%)</i> | <i>Computational Time (Seconds)</i> |
| 0.001 | 100 | 16.250 | 70 | 17.621 |
| 0.0001 | 100 | 18.838 | 75 | 19.935 |

Seperti yang terlihat pada Tabel 4., akurasi tertinggi dalam pengujian nilai *learning rate* adalah 75% dengan waktu komputasi 19,935 detik. Akurasi tersebut didapat dari nilai *learning rate*=0,0001. Hal ini terjadi karena, semakin kecil nilai *learning rate*, semakin kecil juga *step size* dalam pencarian nilai *loss minimum* pada proses pelatihan. Sedangkan, jika nilai *learning rate* terlalu besar, ukuran *step* pencarian nilai *loss minimum* akan semakin besar dan memungkinkan titik optimum terlompati dan hasil performansi sistem akan kurang baik. Oleh sebab itu, hasil pengujian performansi sistem dengan *learning rate*=0,0001 lebih baik dibandingkan dengan sistem yang menggunakan nilai *learning rate*=0,001.

3.4 Pengujian pengaruh SRM

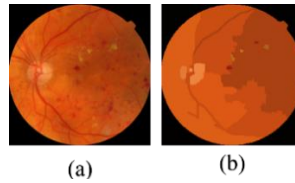
Pengujian ini dilakukan untuk mengetahui pengaruh penggunaan SRM terhadap nilai akurasi dan waktu komputasi. Proses pengujian pada tahap ini membandingkan sistem yang menggunakan metode segmentasi ciri SRM dengan sistem yang tidak menggunakan metode segmentasi SRM. Pada pengujian ini digunakan jumlah *epoch*=100, *learning rate*=0,0001 dan nilai $Q=256$ untuk sistem yang menggunakan segmentasi ciri SRM. Tabel 5. menunjukan hasil pengujian pengaruh perbandingan sistem dengan SRM dan tanpa SRM.

Tabel 5. Pengaruh SRM terhadap Akurasi dan Waktu Komputasi

| <i>SRM</i> | <i>Training Data</i> | | <i>Testing Data</i> | |
|--------------------|----------------------|-------------------------------------|---------------------|-------------------------------------|
| | <i>Accuracy (%)</i> | <i>Computational Time (Seconds)</i> | <i>Accuracy (%)</i> | <i>Computational Time (Seconds)</i> |
| <i>With SRM</i> | 100 | 18.838 | 75 | 19.935 |
| <i>Without SRM</i> | 100 | 18.864 | 62,5 | 20.209 |

Seperti yang terlihat pada Tabel 5., sistem yang menggunakan SRM menghasilkan nilai akurasi yang lebih tinggi dibandingkan sistem yang tidak menggunakan SRM. Sistem dengan SRM memiliki nilai akurasi 75% dengan waktu komputasi 19,935 detik. Hal ini dikarenakan citra yang telah melewati proses segmentasi ciri SRM mempunyai representasi citra yang lebih sederhana dibandingkan citra

yang tidak melewati proses segmentasi ciri SRM. Seperti yang dapat dilihat pada Gambar 11. Oleh sebab itu citra lebih mudah dianalisis oleh sistem.



Gambar 2. Perbandingan Citra (a) Dengan SRM (b) Tanpa SRM

3.5 Pengujian Pengaruh Perbandingan Jumlah Data Latih dan Data Uji

Pengujian ini dilakukan untuk mengetahui pengaruh perbandingan jumlah data latih dan data uji terhadap nilai akurasi dan waktu komputasi. Pada pengujian ini digunakan jumlah $epoch=100$, $learning\ rate=0,0001$ dan nilai $Q=256$. Proses pengujian pada tahap ini membandingkan sistem yang menggunakan perbandingan 1:1 jumlah data latih dan data uji, dengan sistem yang menggunakan perbandingan 3:2 dan 2:3 data latih dan data uji. Tabel 6. menunjukkan pengaruh perbandingan jumlah data latih dan data uji yang berbeda terhadap akurasi dan waktu komputasi.

Tabel 6. Pengaruh Perbandingan Jumlah Data Latih dan Data Uji terhadap Akurasi dan Waktu Komputasi

| Training Data:Testing Data | Training Data | | Testing Data | |
|----------------------------|---------------|------------------------------|--------------|------------------------------|
| | Accuracy (%) | Computational Time (Seconds) | Accuracy (%) | Computational Time (Seconds) |
| 2:3 | 100 | 19.649 | 62.5 | 20.710 |
| 1:1 | 100 | 18.838 | 75 | 20.209 |
| 3:2 | 100 | 21.960 | 81.25 | 14.518 |

Seperti yang terlihat pada Tabel 6, semakin banyak data latih maka akurasi sistem semakin meningkat. Jumlah data latih yang terlalu sedikit akan mempengaruhi proses pembelajaran pada pelatihan jaringan, sehingga hasil yang diperoleh kurang optimal. Nilai akurasi tertinggi diperoleh dari sistem yang menggunakan perbandingan 3:2 data latih dan data uji, yaitu 81,25% dengan waktu komputasi 14,518 detik.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan pada Tugas Akhir ini, sistem yang dirancang dengan metode segmentasi Statistical Region Merging dan metode klasifikasi Convolutional Neural Network mampu mengklasifikasikan tingkat keparahan DR menjadi 4 kelas yaitu normal, mild NPDR, moderate NPDR dan severe NPDR. Nilai akurasi tertinggi yang diperoleh adalah 81,25% dengan waktu komputasi 14,518 detik. Nilai akurasi tersebut diperoleh dengan menggunakan metode segmentasi ciri SRM, dengan perbandingan 3:2 jumlah data latih dan data uji., nilai parameter kompleksitas segmentasi $Q=256$, jumlah $epoch=100$, dan $learning\ rate=0,0001$.

Daftar Pustaka

- [1] World Health Organization, "Blindness and vision impairment prevention." [Online]. Available: <https://www.who.int/blindness/causes/priority/en/index5.html>. [Accessed: 20-Feb-2019].
- [2] "The Prevalence of and Factors Associated With Diabetic Retinopathy in the," vol. 26, no. 6, pp. 1731–1737, 2003.
- [3] R. Y. Dillak and A. Harjoko, "Klasifikasi Fase Retinopati Diabetes Menggunakan Backpropagation Neural Network," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 7, no. 1, pp. 23–34, 2013.
- [4] E. Decencière *et al.*, "Feedback on a publicly distributed image database: The Messidor database," *Image Anal. Stereol.*, vol. 33, no. 3, pp. 231–234, 2014.
- [5] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452–1458, 2004.
- [6] I. W. S. E. P., A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," vol. 5, no. 1, 2016.
- [7] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," *2017 IEEE 2nd Int. Conf. Big Data Anal. ICBDA 2017*, pp. 721–724, 2017.
- [8] R. N. Hidayat, R. R. Isnanto, and O. D. Nurhayati, "Implementasi Jaringan Syaraf Tiruan Perambatan Balik untuk Memprediksi Harga Logam Mulia Emas Menggunakan Algoritma Lavenberg Marquardt," *J. Teknol. dan Sist. Komput.*, vol. 1, no. 2, p. 49, 2016.
- [9] MathWorks, "Transfer Learning Using Alexnet." [Online]. Available: <https://www.mathworks.com/help/deeplearning/examples/transfer-learning-using-alexnet.html>.