

Penerapan Convolutional Neural Network Untuk Klasifikasi Tingkat Keparahan Retinopati Diabetik Pada Penderita Diabetes Melitus

Fithra Hayati Syahrul¹⁾, dan Priyo Sidik Sasongko²⁾

Departemen Ilmu Komputer / Informatika, Universitas Diponegoro

¹⁾fithrahayatisyahrul@students.undip.ac.id, ²⁾priyosidiksasongko@lecturer.undip.ac.id

Abstrak

Retinopati Diabetik adalah penyakit yang dapat mengganggu pembuluh darah retina yang menjadi penyebab kebutaan bagi penderita Diabetes Melitus. Jika penyakit ini terlambat ditangani maka penderita dapat mengalami kebutaan. Perawatan dan pemeriksaan yang tepat dapat membantu mencegah meningkatnya keparahan Retinopati Diabetik. Pemeriksaan secara manual oleh dokter mata dalam mendiagnosis penyakit ini membutuhkan waktu yang relatif lama, sehingga diperlukan sistem untuk mengklasifikasikan tingkat keparahan Retinopati Diabetik. Sistem yang dirancang pada penelitian ini menggunakan metode Convolutional Neural Network untuk klasifikasi tingkat keparahan Retinopati Diabetik. Tingkat keparahan Retinopati Diabetik dibagi menjadi 5 kelas yaitu NO DR, Mild, Moderate, Severe, dan Proliferative DR. Penelitian Penerapan Convolutional Neural Network untuk Klasifikasi Tingkat Keparahan Retinopati Diabetik pada Penderita Diabetes Melitus menggunakan citra berukuran 64 x 64 x 3 dengan channel RGB. Tahap pra-pengolahan citra yang dilakukan adalah pengubahan ukuran citra. Arsitektur CNN yang digunakan terdiri dari 5 blok dimana masing-masing blok berisi batch normalization layer, convolution layer, max pooling layer menggunakan parameter learning rate 0.0005. Hasil evaluasi model 652 data uji menunjukkan akurasi terbaik sebesar 91.10%.

Kata kunci : Retina, Retinopati diabetik, Convolutional Neural Network

Abstract

Diabetic retinopathy is a disease that can interfere with retinal blood vessels that cause blindness for people with diabetes mellitus. If the disease is treated too late then the patient can experience blindness. Proper care and examination can help prevent the increasing severity of diabetic retinopathy. Manual examination by an ophthalmologist in diagnosing this disease requires a relatively long time, so we need a system to classify the severity of diabetic retinopathy. The system designed in this study uses the Convolutional Neural Network method to classify the severity of diabetic retinopathy. The severity of Diabetic Retinopathy is divided into 5 classes namely NO DR, Mild, Moderate, Severe, and Proliferative DR. Research on the Application of Convolutional Neural Network to Classify the Severity of Diabetic Retinopathy in Diabetes Mellitus Patients using 64 x 64 x 3 images with RGB channels. The image pre-processing stage is done by resizing the image. The CNN architecture used consists of 5 blocks where each block contains a batch normalization layer, convolution layer, max pooling layer using the learning rate parameter of 0.0005. The results of the evaluation of the 652 model test data showed the best accuracy of 91.10%.

Keywords : Retina, Retinopati diabetic, Convolutional Neural Network

1 PENDAHULUAN

Diabetes Melitus (DM) adalah penyakit kronik yang banyak diderita di dunia. Jumlah populasi yang mengalami penyakit ini mencapai 2.8% atau sebanyak 171 juta penderita di dunia pada tahun 2000. Indonesia menempati urutan ke-4 di dunia setelah India, Cina, dan Amerika Serikat sebagai negara dengan penderita DM sebesar 8.4 juta pada tahun 2000 serta diprediksi akan meningkat menjadi 21.3 juta penderita pada tahun 2030 [1].

Retinopati diabetik (RD) merupakan komplikasi DM pada mata yang menyebabkan kebutaan terbanyak setelah katarak, hal ini meningkat seiring dengan meningkatnya jumlah penderita DM [2]. Menurut WHO kebutaan paling banyak terjadi karena katarak (47.8%), *glaucoma* (12.3%), *uveitis* (10.2%), *age macular degeneration* (AMD) (8.7%), *trachoma* (3.6%), *corneal opacity* (5.1%), dan *diabetic retinopathy* (4.8%) [3]. Kebutuan yang diakibatkan oleh RD dapat menurunkan kualitas hidup dan produktivitas penderita yang menimbulkan masalah sosial, sehingga penyakit ini menjadi masalah kesehatan yang diwaspadai di dunia, hal ini disebabkan keterlambatan diagnosis oleh dokter spesialis mata karena pada tahap awal tidak mengalami gangguan penglihatan. Diagnosis RD secara dini dapat membantu perawatan medis dan prognosis. Diagnosis otomatis berdasarkan pencitraan medis merupakan bagian penting dari perawatan menggunakan teknologi, beberapa sistem diagnosis dibantu komputer telah diusulkan untuk membantu dokter mata dalam mengenali RD.

2 TINJAUAN PUSTAKA

Penelitian terkait klasifikasi penyakit RD telah dilakukan menggunakan berbagai metode seperti pada penelitian tentang *Wavelet Haar* dan *Backpropagation Neural Network* untuk klasifikasi Retinopati Diabetik. Proses yang dilakukan adalah *preprocessing*, ekstraksi ciri menggunakan *wavelet haar* pada level 1 dan 4, normalisasi dan klasifikasi. Hasilnya tingkat akurasi terendah pada level 1 mencapai 45.16% dan data uji 20%, sedangkan tingkat akurasi tertinggi sebesar 55% pada ukuran citra 2240 x 1488 piksel dan 1440 x 960 piksel dengan data uji 10% lalu tingkat akurasi terendah pada level 4 adalah 40.63% dengan ukuran citra 2304 x 1536 piksel dan 1440 x 960 piksel dan data uji 5% sedangkan tingkat akurasi tertinggi sebesar 56.25% pada ukuran citra 2240 x 1488 piksel dengan data uji 5% [4]. Kemudian penelitian lain tentang sistem klasifikasi tingkat keparahan Retinopati Diabetik menggunakan *Support Vector Machine*. Proses yang dilakukan adalah akuisisi citra, pra-pengolahan citra, deteksi, ekstraksi ciri, dan klasifikasi. Evaluasi dari klasifikasi menggunakan parameter akurasi (*accuracy*), kekhususan (*specificity*), dan sensitivitas (*sensitivity*). Hasil penelitian klasifikasi pada 240 data latih dan 60 data uji menghasilkan nilai rata-rata akurasi sebesar 95.93%, *specificity* 97.29% dan *sensitivity* 91.07% [5].

Selanjutnya penelitian untuk deteksi dan klasifikasi Retinopati Diabetik menggunakan CNN untuk klasifikasi menjadi 4 kelas yaitu normal, edema makula, diabetes proliferasif retinopati, dan retinopati diabetes non-proliferasif. Model CNN penelitian ini memiliki delapan lapisan yaitu lapisan input gambar, lapisan konvolusi, lapisan ReLU, lapisan normalisasi, lapisan *max pooling*, lapisan *fully connected*, lapisan *softmax*, dan klasifikasi. Hasil evaluasi kinerja yang diperoleh akurasi 97%, sensitivitas 94%, spesifisitas 98%, precision 94% [6]. Lalu, penelitian klasifikasi Retinopati Diabetik menggunakan arsitektur *deep learning* untuk

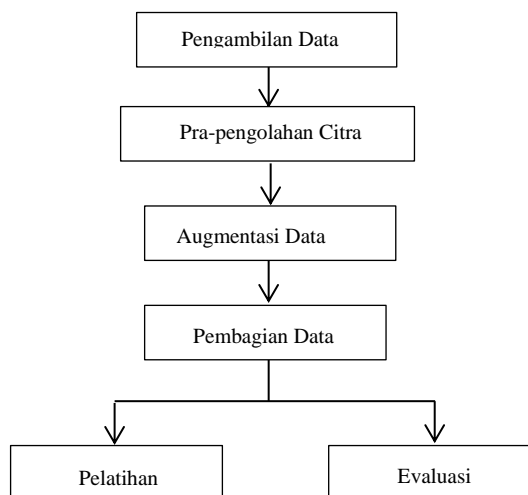
klasifikasi menjadi 5 kelas yaitu normal, *mild*, *moderate*, *severe non-proliferative*, dan *proliferative diabetic retinopathy*. Penelitian ini menggunakan citra berukuran 1x256x256, kemudian melakukan proses augmentasi yaitu *contrast adjustment*, *flipping images*, dan *brightness adjustments*. Setelah itu, perubahan ukuran citra, konversi citra ke *grayscale*, dan konversi ke model L. Arsitektur CNN yang digunakan terdiri dari lapisan konvolusi, lapisan *pooling*, lapisan ReLU, lapisan *dropout*, lapisan *fully connected*, dan lapisan klasifikasi. Hasil performa yang diperoleh sebesar 94% [7].

Pendekatan deep learning telah meningkatkan kinerja prediksi dalam klasifikasi dan diagnosis penyakit berdasarkan pencitraan medis. Pendekatan diterapkan tidak hanya untuk memprediksi keberadaan penyakit tetapi juga untuk membedakan kelas penyakit [8]. Penelitian yang dilakukan oleh Jude Hemanth D., Omer Deperlioglu, dan Utku Kose dalam mendeteksi dan klasifikasi Retinopati Diabetik menggunakan CNN untuk 4 kelas, menghasilkan evaluasi kinerja yang sangat baik dibuktikan dengan nilai akurasi yang diperoleh sebesar 97%, lalu sensitivitas 94%, spesifisitas 98%, dan precision 94%.

Penelitian ini menerapkan metode CNN untuk mengklasifikasi tingkat keparahan Retinopati Diabetik pada penderita Diabetes Melitus. Penelitian ini menggunakan citra fundus yang diklasifikasikan menjadi 5 kelas yaitu *NO DR*, *Mild*, *Moderate*, *Severe*, dan *Prolifetaive DR*. Harapannya penelitian ini dapat membantu dokter mata dalam mendiagnosis Retinopati Diabetik.

3 METODE PENELITIAN

Penelitian klasifikasi tingkat keparahan Retinopati Diabetik ini diperlukan suatu langkah penyelesaian masalah yang sistematis untuk mendapatkan hasil yang tepat. Langkah penyelesaian masalah ditunjukkan pada Gambar 1.



Gambar 1 Garis Besar Penyelesaian Masalah

3.1 PENGAMBILAN DATA

Pengambilan data pada penelitian ini dari APTOS 2019 *Blindness Detection kaggle*, secara keseluruhan terdapat 3662 citra menggunakan channel RGB yang terbagi menjadi 5

kelas yaitu 0 (*NO DR / Normal*), 1 (*Mild / Retinopati Nonproliferatif Ringan*), 2 (*Moderate / Retinopati Nonproliferatif Sedang*), 3 (*Severe / Retinopati Nonproliferatif Berat*), dan 4 (*Proliferative DR / Retinopati Diabetik Proliferatif*).

3.2 PRA-PENGOLAHAN CITRA

Pra-pengolahan citra berguna untuk mempersiapkan data dari bentuk data mentah menjadi data yang siap digunakan pada tahapan selanjutnya, pada penelitian ini input berupa gambar .jpg atau .png berbagai ukuran misalnya 2416x1736x3 kemudian ukuran setiap gambar diubah menjadi 64x64x3. Kemudian gambar-gambar tersebut diubah menjadi .csv. Channel warna yang digunakan adalah *Red, Green, Blue* (RGB).

3.3 AUGMENTASI DATA

Augmentasi data sangat penting untuk dilakukan untuk menyeimbangkan jumlah data citra agar mencapai kinerja yang memuaskan. Penelitian ini menggunakan augmentasi data yaitu *rotation image*, dan *flipping image*.

3.4 PEMBAGIAN DATA

Proses pembagian data yaitu memisahkan data latih untuk pelatihan dan data uji untuk pengujian. Perbandingan data latih dan data uji adalah 9:1 dimana 5860 data yang masuk ke proses pelatihan dan 652 data yang digunakan untuk proses pengujian. Pembagian data menggunakan parameter seed untuk memastikan angka random yang dihasilkan benar-benar acak.

3.5 ARSITEKTUR CNN

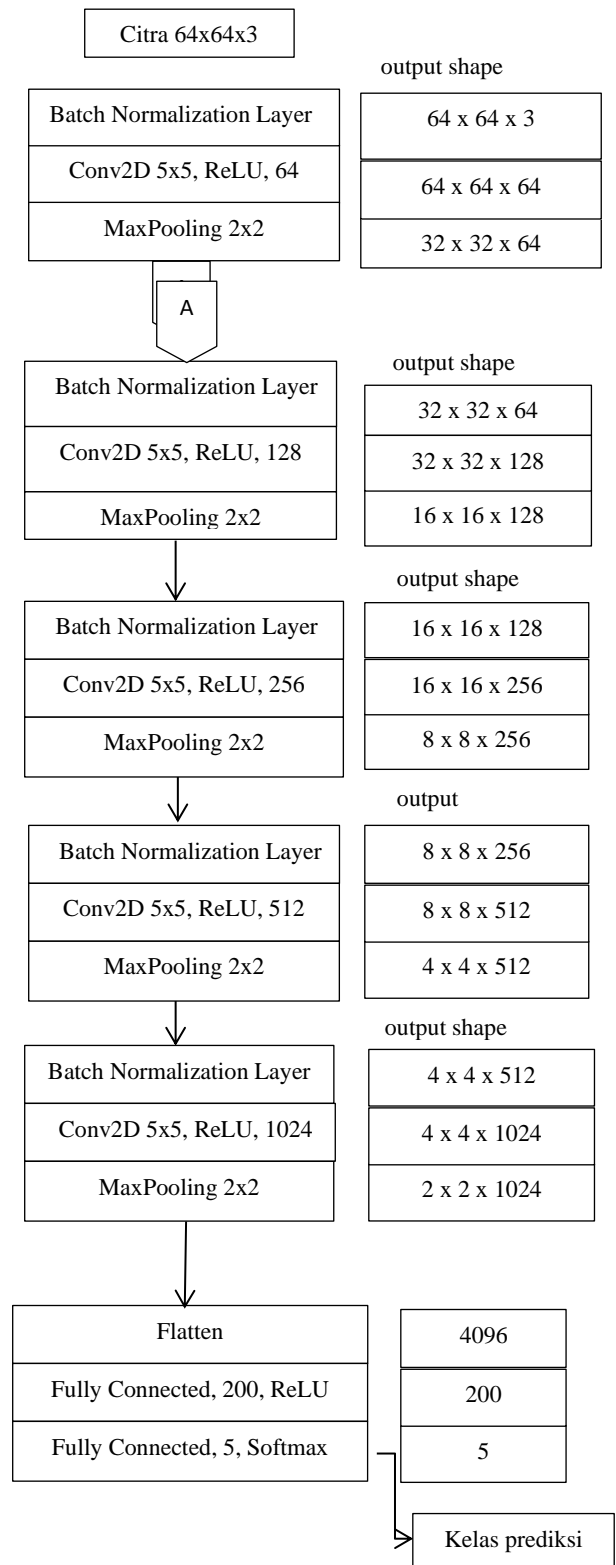
CNN adalah algoritma untuk identifikasi citra berdasarkan kelas-kelas. Pelatihan pada CNN dilakukan pada data-data yang telah diberi label sebagai data latih dengan melakukan konvolusi citra untuk mendapatkan fitur yang dapat digunakan pada fully connected layer untuk diarahkan kepada kelas tertentu.

Penelitian ini merancang arsitektur yang terdiri dari 1 batch normalization layer dengan ukuran kernel 5x5 pada 1 layer konvolusi menggunakan aktivasi ReLU, dan 1 layer Max pooling. Blok layer ini diulang sebanyak 5 kali untuk setiap citra input. Setiap blok dilengkapi 1 *batch normalization layer* untuk menormalkan nilai citra masukan, sedangkan pada akhir blok terdapat layer *Max pooling* untuk mencari nilai maksimum pada setiap dimensi vektor yang berguna sebagai output untuk tahap selanjutnya. Input masukan citra ukuran 64 x 64 x 3 (panjang x lebar x channel), dapat dilihat pada Gambar 2.

3.6 PELATIHAN MODEL

Data latih yang digunakan untuk pelatihan sebanyak 5.860 citra. Data latih memiliki variabel label terpisah dengan representasi *categorical*. *Optimizer* pelatihan untuk tiap CNN yang digunakan adalah Adam. Pelatihan pada tiap CNN menggunakan epoch sebanyak 100. Pengukuran pengujian sementara (validasi) pada proses pelatihan di tiap epoch berupa *error (loss)* dan akurasi. Pengukuran *loss* yang digunakan adalah *categorical cross entropy* yaitu pengukuran *error* untuk kasus *multiclass*. Paramater EarlyStopping dengan *patience* sebanyak 100, artinya pelatihan akan berhenti saat terjadi peningkatan sebanyak 100 kali pada *loss*

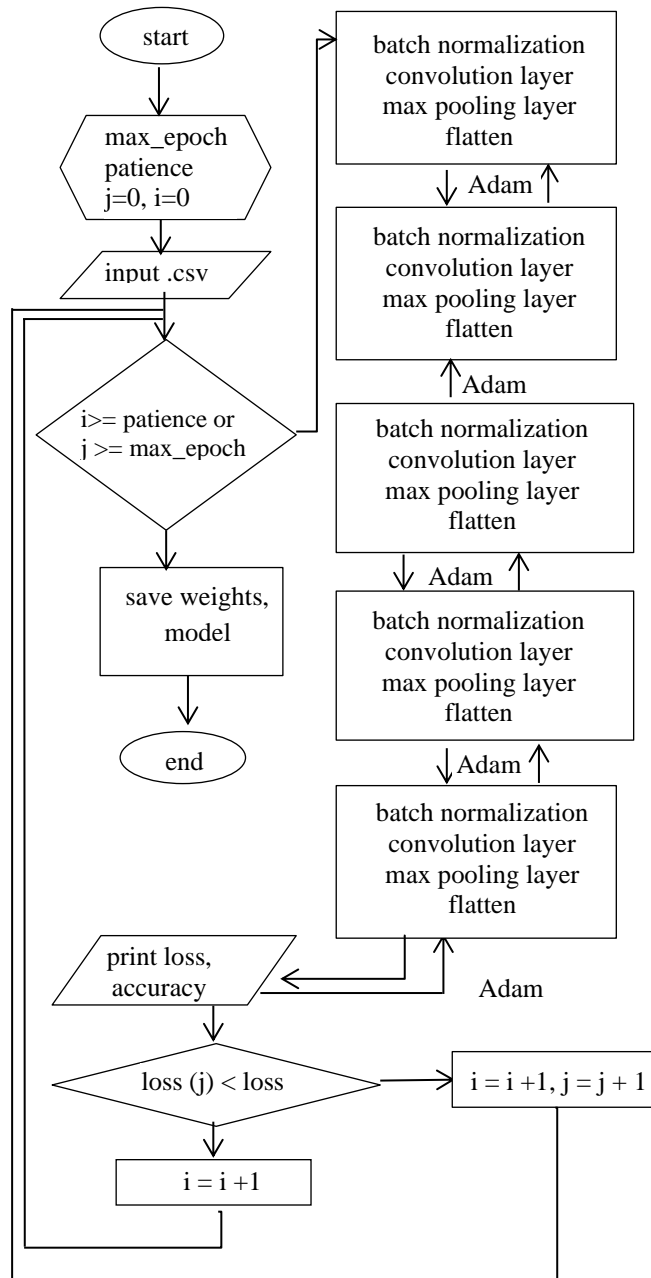
dengan tujuan menghindari *overfitting* lebih dini, sehingga pelatihan tiap CNN akan berhenti saat *patience* mencapai 100.



Gambar 2 Arsitektur CNN Penelitian

Pelatihan dilakukan secara berulang sebanyak lima blok dimana setiap blok terdiri dari proses *batch normalization*, *convolution layer*, dan *max pooling layer* hingga menghasilkan nilai akurasi yang optimal dan mencegah terjadinya *underfit* (keadaan di mana model data latih yang digunakan tidak mewakili seluruh data yang digunakan) dan *overfit* pada data. *Overfitting* dihindari dengan penggunaan parameter *early stopping*. Proses pelatihan CNN dapat dilihat pada Gambar 3.

3.7 PROSES FORWARD PROPAGATION



Gambar 3 Flowchart Pelatihan CNN

Proses pelatihan terjadi dari blok awal sampai blok akhir yang berisi layer *batch normalization*, konvolusi, dan *max pooling* dengan memisalkan input sebuah channel.

Langkah-langkah perhitungan dibagi menjadi 3 bagian yaitu bagian *forward propagation*, *back propagation*, dan *update parameter*.

Menurut [9], proses *feed-forward* pada CNN adalah sebagai berikut :

- 1) Inisialisasi bias dengan nilai 0, serta nilai awal bobot pada lapisan *fully connected* dan lapisan konvolusi secara acak.

- 2) Proses konvolusi.

$$C_p^q = f(I * k_{1,p}^q + b_p^q) \quad (1)$$

- 3) Proses subsampling oleh lapisan pooling.

$$S_p^q(i, j) = \max(\sum_{u=0}^1 \sum_{v=0}^1 C_p^q(i, j)) \quad (2)$$

- 4) Proses vectorization dan concatenation.

$$fc = Fc(\{S_p^q\}) \quad (3)$$

- 5) Menghitung prediksi target fitur yang masuk ke dalam lapisan fully connected.

$$\hat{y}(i) = f(\sum_{j=1}^n W(i, j) \cdot fc(j) + b(i)) \quad (4)$$

- 6) Menghitung nilai loss.

$$L(y^{L+1}, y) = -\sum_{j=1}^n y_j \log(y_j^{L+1}) \quad (5)$$

3.8 PROSES BACKPROPAGATION

Proses backpropagation pada CNN menurut [9] adalah sebagai berikut :

- 1) Perhitungan perubahan bobot W yang terhubung dengan neuron penghasil fitur dari lapisan hidden.

$$\Delta W(i, j) = (\hat{y}(i) - y(i)) \cdot \hat{y}(i)(1 - \hat{y}(i)) \cdot fc(j) \quad (6)$$

- 2) Update bias.

$$\Delta b(i) = (\hat{y}(i) - y(i)) \cdot \hat{y}(i)(1 - \hat{y}(i)) \quad (7)$$

- 3) Menghitung perubahan feature maps.

$$\Delta fc(j) = \sum_{i=1}^n \Delta \hat{y}(i) \cdot W(i, j) \quad (8)$$

- 4) Mengubah vektor panjang ke bentuk matriks 2 dimensi.

$$\{\Delta S_p^q\}_{p=1,2,\dots,n} = F^{-1}(\Delta fc) \quad (9)$$

- 5) Proses upsampling.

$$\Delta C_p^q(i, j) = (\Delta S_p^q(\left\lfloor \frac{i}{2} \right\rfloor, \left\lfloor \frac{j}{2} \right\rfloor)), p = 1, 2, \dots, n \quad (10)$$

- 7) Menghitung perubahan nilai filter.

$$\Delta k_{1,p}^q = I_{rot180} * \Delta C'_{p,f}^q \quad (11)$$

8) Menghitung perubahan nilai bias.

$$\Delta b_p^q = \sum_{i=1}^n \sum_{j=1}^n \Delta C_{p,f}^q(i, j) \quad (12)$$

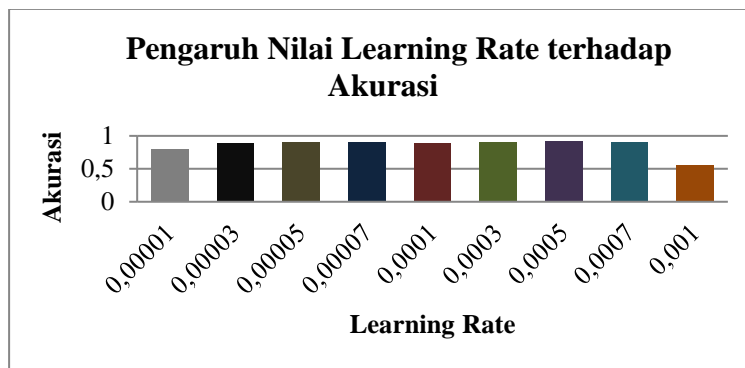
9) Update parameter.

4 HASIL DAN PEMBAHASAN

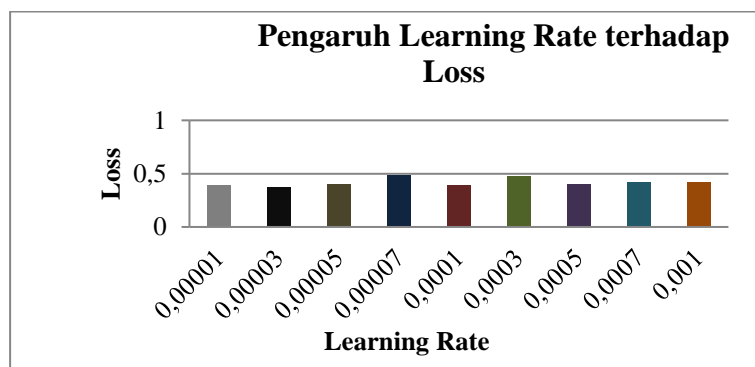
Penelitian ini terdapat enam skenario pelatihan model, yaitu pertama untuk mengetahui pengaruh perubahan parameter learning rate yaitu 0.00001, 0.00003, 0.00005, 0.00007, 0.0001, 0.0003, 0.0005, 0.0007, dan 0.001 pada saat pelatihan, kedua pengaruh penggunaan jenis pooling layer pada model saat pelatihan yaitu *Max Pooling Layer* dan *Average Pooling Layer*, ketiga pengaruh penggunaan *dropout* dan tanpa menggunakan *dropout* pada model pelatihan, keempat pengaruh ukuran kernel sebesar 3x3, 5x5, dan 7x7 pada model pelatihan, kelima pengaruh jumlah blok pada model saat pelatihan yaitu 3 blok, 4 blok, dan 5 blok, lalu keenam pengaruh penggunaan *batch normalization* dan tidak menggunakan *batch normalization* pada model pelatihan.

4.1.1 HASIL SKENARIO 1

Hasil skenario 1 menampilkan grafik pengaruh nilai *learning rate* terhadap akurasi yang dapat dilihat pada Gambar 4 dan grafik pengaruh nilai *learning rate* terhadap *loss* dapat dilihat pada Gambar 5.



Gambar 4 Grafik Pengaruh Nilai *Learning Rate* terhadap Akurasi

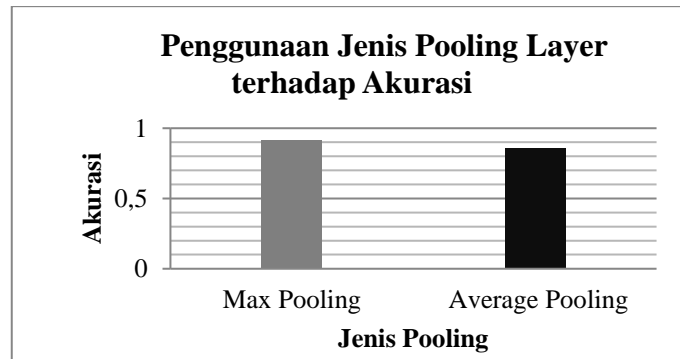


Gambar 5 Grafik Pengaruh Nilai *Learning Rate* terhadap Loss

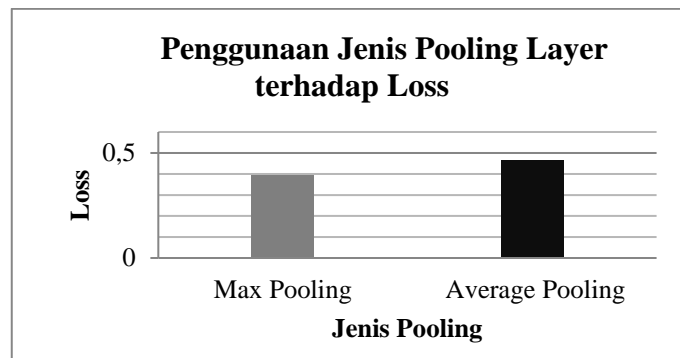
Hasil percobaan dengan perubahan nilai parameter learning rate menunjukkan bahwa learning rate yang bernilai 0.0005 menghasilkan performa akurasi dan loss paling baik. Akurasi yang diperoleh sebesar 0.9110 dan nilai loss 0.3944.

4.1.2 HASIL SKENARIO 2

Hasil skenario 2 menampilkan penggunaan jenis *pooling layer* terhadap akurasi dapat dilihat pada Gambar 6 dan grafik penggunaan jenis *pooling layer* terhadap loss dapat dilihat pada Gambar 7.



Gambar 6 Grafik Penggunaan Jenis *Pooling Layer* terhadap Akurasi

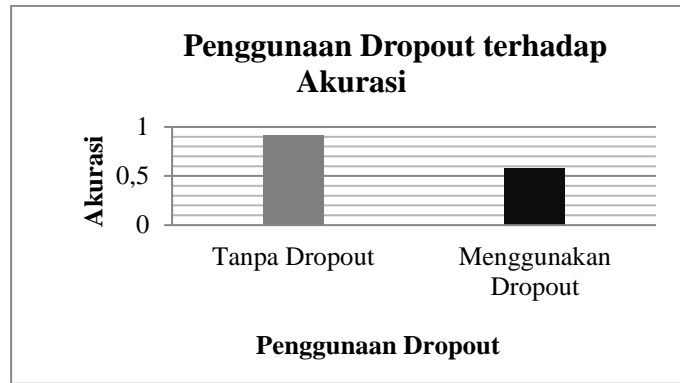


Gambar 7 Grafik Penggunaan Jenis *Pooling Layer* terhadap Loss

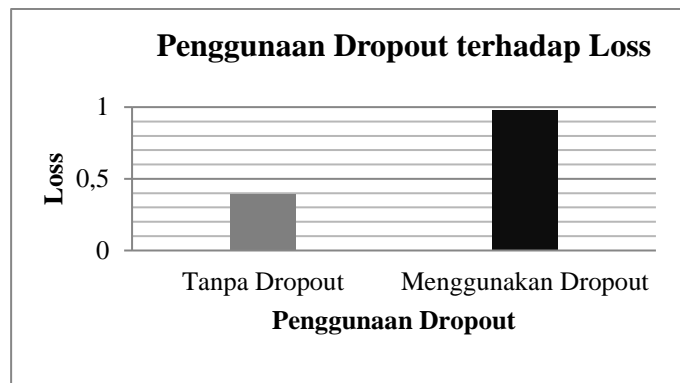
Hasil percobaan penggunaan jenis *pooling layer* terhadap akurasi dan *loss* menunjukkan bahwa penggunaan *Max Pooling Layer* menghasilkan performa akurasi dan *loss* paling baik, dibuktikan dengan nilai akurasi yang lebih baik yaitu sebesar 0.9110 dan nilai *loss* yang lebih kecil sebesar 0.3944 dibandingkan dengan penggunaan *Average Pooling Layer* yang menghasilkan nilai akurasi sebesar 0.8580 dan nilai *loss* 0.4650.

4.1.3 HASIL SKENARIO 3

Hasil skenario 3 menampilkan pengaruh penggunaan *dropout* dan tanpa menggunakan *dropout* terhadap akurasi dapat dilihat pada Gambar 8 dan grafik pengaruh *dropout* dan tanpa menggunakan *dropout* terhadap *loss* dapat dilihat pada Gambar 9.



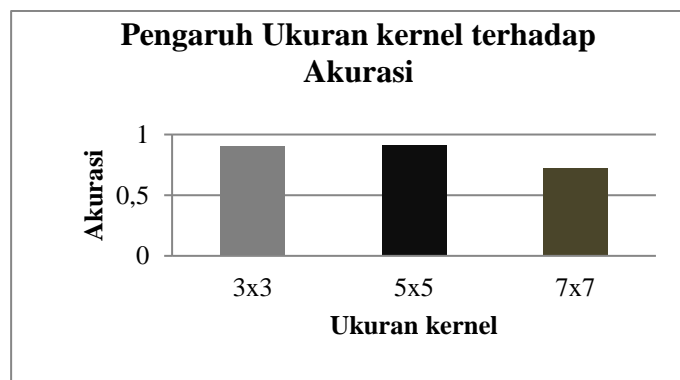
Gambar 8 Grafik Pengaruh Dropout terhadap Akurasi



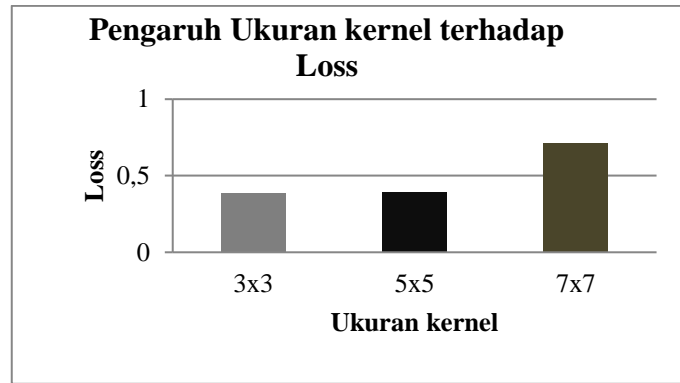
Gambar 9 Grafik Pengaruh Dropout terhadap Loss

4.1.4 HASIL SKENARIO 4

Hasil skenario 4 menampilkan pengaruh ukuran kernel terhadap akurasi dapat dilihat pada Gambar 10 dan grafik pengaruh ukuran kernel terhadap loss dapat dilihat pada Gambar 11. Hasil percobaan pengaruh ukuran kernel terhadap akurasi dan loss menunjukkan bahwa menggunakan ukuran kernel 5x5 menghasilkan performa akurasi dan loss paling baik, nilai akurasi sebesar 0.9110 dan nilai loss sebesar 0.3944 dibandingkan dengan ukuran kernel 3x3 yang menghasilkan nilai akurasi sebesar 0.9043 dan nilai loss 0.3827 serta ukuran kernel 7x7 menghasilkan nilai akurasi sebesar 0.7191 dan nilai loss 0.7155.



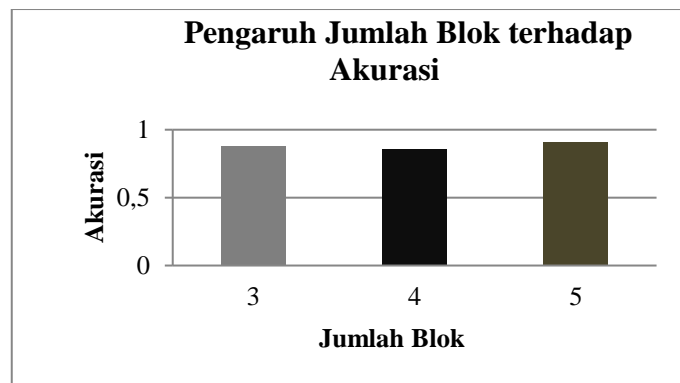
Gambar 10 Grafik Pengaruh Ukuran kernel terhadap Akurasi



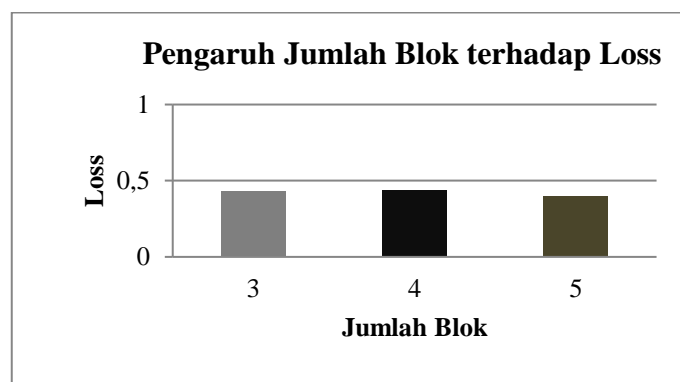
Gambar 11 Grafik Pengaruh Ukuran kernel terhadap Loss

4.1.5 HASIL SKENARIO 5

Hasil skenario 5 menampilkan pengaruh jumlah blok terhadap akurasi dapat dilihat pada Gambar 12 dan grafik pengaruh jumlah blok terhadap loss dapat dilihat pada Gambar 13.



Gambar 12 Grafik Pengaruh Jumlah Blok terhadap Akurasi

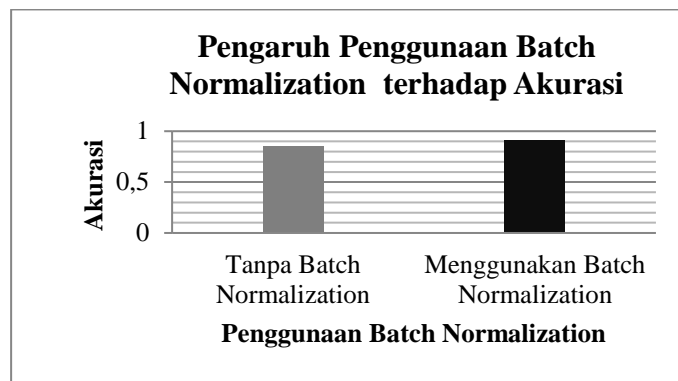


Gambar 13 Grafik Pengaruh Jumlah Blok terhadap Loss

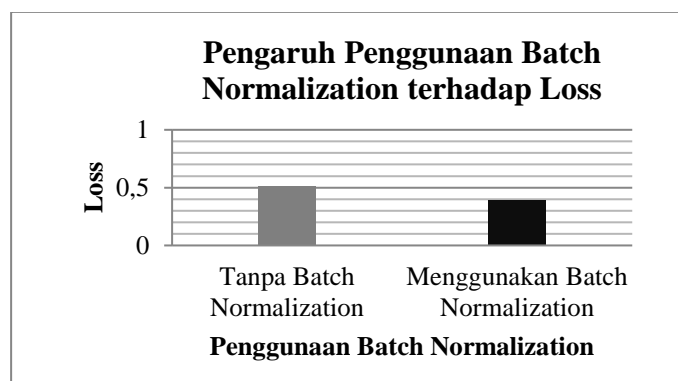
Hasil percobaan pengaruh ukuran kernel terhadap akurasi dan loss menunjukkan bahwa menggunakan jumlah blok 5 menghasilkan performa akurasi dan loss paling baik, dengan nilai akurasi sebesar 0.9110 dan nilai loss sebesar 0.3944 dibandingkan dengan jumlah blok 3 yang menghasilkan nilai akurasi sebesar 0.9043 dan nilai loss 0.3827 serta jumlah blok 4 yang menghasilkan nilai akurasi sebesar 0.8573 dan nilai loss 0.4337.

4.1.6 HASIL SKENARIO 6

Hasil skenario 6 menampilkan pengaruh penggunaan *batch normalization* terhadap akurasi dapat dilihat pada Gambar 14 dan grafik pengaruh penggunaan *batch normalization* terhadap *loss* dapat dilihat pada Gambar 15.



Gambar 14 Grafik Pengaruh Penggunaan Batch Normalization terhadap Akurasi

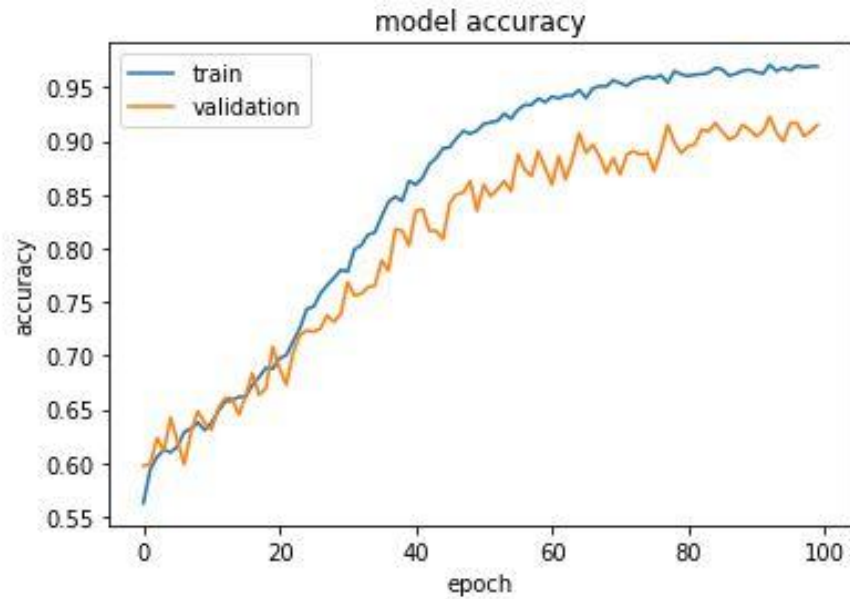


Gambar 15 Grafik Pengaruh Penggunaan Batch Normalization terhadap Loss

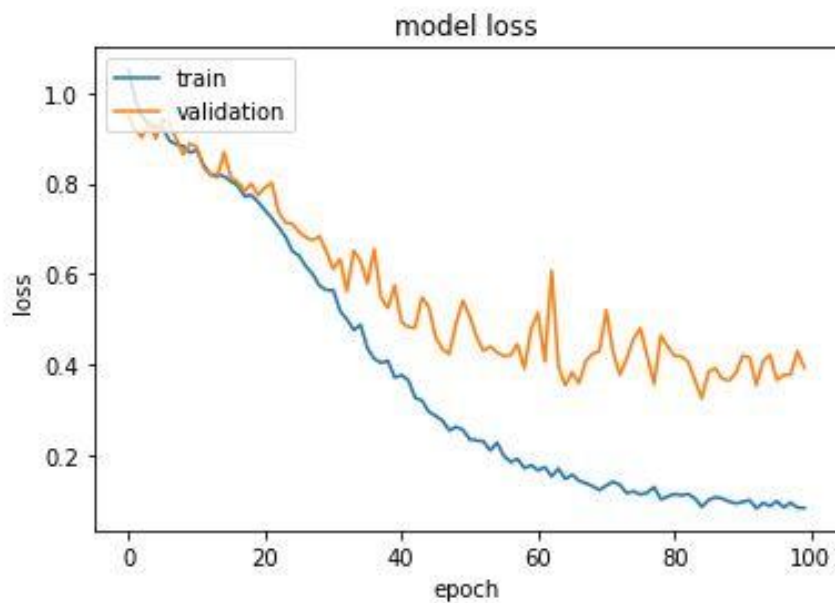
Hasil percobaan pengaruh penggunaan *batch normalization* terhadap akurasi dan *loss* menunjukkan bahwa menggunakan *batch normalization* menghasilkan performa akurasi dan *loss* lebih baik, dibuktikan dengan nilai akurasi sebesar 0.9110 dan nilai *loss* sebesar 0.3944 dibandingkan dengan tanpa menggunakan *batch normalization* yang menghasilkan nilai akurasi sebesar 0.8457 dan nilai *loss* 0.5132.

4.2 ANALISA HASIL SEMUA SKENARIO

Hasil skenario terbaik dari hasil skenario penelitian yang sudah dilakukan untuk klasifikasi tingkat keparahan Retinopati Diabetik pada penderita Diabetes Melitus adalah pada nilai parameter learning rate 0.0005, menggunakan *Max Pooling Layer*, tanpa menggunakan *dropout*, menggunakan kernel 5x5, jumlah blok yang digunakan sebanyak 5 blok, serta menggunakan *batch normalization* yang dibuktikan dengan hasil akurasi yang paling baik dan nilai *loss* paling kecil dibandingkan lainnya. Grafik pengaruh epoch terhadap akurasi pada hasil skenario terbaik dapat dilihat pada Gambar 16 serta grafik pengaruh epoch terhadap *loss* dapat dilihat pada Gambar 17.



Gambar 16 Grafik Pengaruh Epoch terhadap Akurasi Hasil Skenario Terbaik



Gambar 17 Grafik Pengaruh Epoch terhadap Loss Hasil Skenario Terbaik

5 KESIMPULAN

Kesimpulan dari hasil penelitian yang dilakukan mengenai penerapan *Convolutional Neural Network* untuk klasifikasi tingkat keparahan retinopati diabetik pada penderita diabetes melitus adalah sebagai berikut:

1. Parameter pada arsitektur CNN yang diujicobakan dengan nilai learning rate sebesar 0.0005 menghasilkan nilai akurasi terbaik sebesar 91.10%.
2. Jenis pooling layer terbaik pada learning rate 0.0005 adalah *Max Pooling Layer* yang menghasilkan performa akurasi dan loss paling baik, dibuktikan dengan nilai akurasi sebesar 91.10%.

3. Arsitektur CNN yang diujicobakan tanpa penggunaan dropout, menggunakan ukuran kernel 5x5 dengan jumlah blok 5 serta menggunakan *batch normalization* menghasilkan akurasi terbaik sebesar 91.10%.
4. Nilai *precision*, *recall*, dan *f1-score* terbaik adalah kelas *NO DR* yang menunjukkan bahwa klasifikasi tingkat keparahan retinopati diabetik pada kelas normal sudah sangat baik. Nilai *precision* yang diperoleh sebesar 0.95 lalu *recall* 0.99, dan *f1-score* 0.97 atau 97% .

DAFTAR PUSTAKA

- [1] S. Wild, G. Roglic, A. Green, R. Sicree dan H. King, "Global Prevalence of Diabetes Estimates for the year 2000 and projections for 2030," *Diabetes Care*, pp. 27(5): 1047-1053, 2004.
- [2] Sitompul, "Retinopati Diabetik," *J Indon Med Assoc*, pp. 61(8) p.337-341., 2011.
- [3] WHO, "Global Data on Visual Impairment," 2002. [Online]. Available: goliath.ecnnext.com/com2/gi_0199-35322637/Global-data-on-visual-impairment.html. [Diakses Agustus 2019].
- [4] S. Sanjaya, A. M. Priyatno, F. Yanto dan I. Afrianty, "Klasifikasi Diabetik Retinopati Menggunakan Wavelet Haar dan Backpropagation Neural Network," *ResearchGate*, pp. 77-84, 2018.
- [5] T. G. A. Putranto dan I. Candradewi, "Sistem Klasifikasi Tingkat Keparahan Retinopati Diabetik Menggunakan Support Vector Machine," *Indonesia Journal of Electronics and Instrumentation Systems (IJEIS)*, pp. 37-48, 2018.
- [6] J. Hemanth, O. Deperlioglu dan U. Kose, "An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network," *Springer*, 2018.
- [7] T. Chandrakumar dan R. Kathirvel, "Classifying Diabetic Retinopathy using Deep Learning Architecture," *International Journal of Engineering Research & Technology (IJERT)*, p. Vol. 5, 2016.
- [8] S. J. Pan dan Q. Y. Fellow, "A Survey on Transfer Learning," *IEEE Transactions on knowledge and data engineering* 22 (10), pp. 1345-1359, 2010.
- [9] Z. Zhang, "Derivation of Backpropagation in Convolutional Neural Network," University of Tennessee, Knoxville, 2016.