

Optimasi *Convolutional Neural Network* dan *K-Fold Cross Validation* pada Sistem Klasifikasi Glaukoma

**YUNENDAH NUR FUADAH, IBNU DAWAN UBAIDULLAH, NUR IBRAHIM,
FAUZI FRAHMA TALININGSING, NIDAAN KHOFIYA SY, MUHAMMAD ADNAN
PRAMUDITHO**

Fakultas Teknik Elektro Universitas Telkom, Indonesia

Email: yunendah@telkomuniversity.ac.id

Received 15 Juni 2022 | Revised 4 Juli 2022 | Accepted 6 Juli 2022

ABSTRAK

Pada penelitian ini dilakukan perancangan arsitektur Convolutional Neural Network (CNN) yang terdiri dari 5 layer konvolusi dan 1-fully connected layer untuk mengklasifikasikan citra fundus kedalam kondisi normal, early, moderate, deep, dan ocular hypertension (OHT). Selanjutnya, model yang diusulkan dibandingkan dengan arsitektur AlexNet yang memiliki 5 layer konvolusi dan 3- fully connected layer. Data yang digunakan berupa citra fundus yang terdiri dari 3200 data latih, 800 data validasi, dan 1000 data uji. Optimasi model CNN dilakukan dengan melakukan pengujian hyperparameter yang terdiri dari learning rate, batch-size, epoch, dan optimizer. Selain itu, pada tahap training diimplementasikan 5-fold cross validation untuk seleksi model terbaik. Dengan model yang lebih sederhana dari AlexNet, model CNN usulan dapat memberikan performansi yang sama dengan arsitektur AlexNet yaitu akurasi 100%, presisi, recall, f1-score dan AUC score bernilai 1.

Kata kunci: glaukoma, citra fundus, convolutional neural network (CNN), AlexNet

ABSTRACT

This study proposes a Convolutional Neural Network with 5 convolutional layer and 1-fully connected layer to classify fundus images into normal, early, moderate, deep, and ocular hypertension (OHT) conditions. Furthermore, the proposed model is compared with the AlexNet architecture which has 5 convolution layers and 3- fully connected layers. The data used is a fundus image consisting of 3200 training data, 800 validation data, and 1000 test data. The optimization of the CNN model is performed by testing the hyperparameters consisting of learning rate, batch size, epoch, and optimizer. In addition, at the training stage, 5-fold cross validation is implemented to select the best model to be used in the test stage. With a simpler model from AlexNet, the proposed model provides 100% accuracy performance with precision values, recall, f1-score, and AUC score of 1.

Keywords: glaucoma, fundus images, convolutional neural network (CNN), AlexNet

1. PENDAHULUAN

Glaukoma merupakan penyakit mata *neurodegeneratif* yang ditandai dengan kerusakan saraf optik yang diakibatkan oleh penekanan bola mata. Kerusakan saraf optik tersebut dapat menyebabkan kebutaan permanen apabila tidak segera mendapatkan penanganan medis yang tepat. Berdasarkan data dari *global prevalence of primary open-angle glaucoma* (POAG) dan *primary angle-closure glaucoma* (PACG) (**Tham, dkk, 2014**), diperkirakan pada tahun 2040 glaukoma akan mempengaruhi sekitar 111,8 juta orang di seluruh dunia. Akibatnya, pemeriksaan mata secara teratur diperlukan dan disarankan untuk deteksi dini glaukoma, mengingat jumlah pasien meningkat secara dramatis seiring bertambahnya usia.

Beberapa penelitian telah mengusulkan algoritma untuk diagnosis glaukoma berbasis *machine learning* dan *deep learning*. (**An, dkk, 2019**) mengembangkan algoritma berbasis *machine learning* untuk diagnosis glaukoma menggunakan berbagai model klasifikasi seperti *neural network* (NN), *naive Bayes* (NB), *support vector machine* (SVM), dan *gradient boosted decision trees* (GBDT). Hasil riset mereka menunjukkan bahwa NN memiliki kinerja klasifikasi terbaik, dengan memperoleh akurasi sebesar 87,8%. Selanjutnya, mereka juga melakukan penelitian untuk diagnosis glaukoma pada pasien berdasarkan citra *three-dimensional optical coherence tomography* (OCT) dan citra fundus menggunakan metode *transfer learning convolutional neural network* (CNN) dan *random forest* (RF) untuk mengklasifikasikan kondisi normal dan glaukoma. Kombinasi model RF dan CNN memberikan nilai Area Under Curve (AUC) 0,963. (**Wu, dkk, 2021**) mengembangkan sistem untuk diagnosis glaukoma berdasarkan citra OCT. Mereka membandingkan lima model klasifikasi yaitu *conditional inference trees* (CIT), *logistic model tree* (LMT), *decision tree*, *random forest* (RF), dan *extreme gradient boosting* (XGBoost). Pada penelitian tersebut, random forest juga memberikan performansi terbaik dibanding dengan model lainnya.

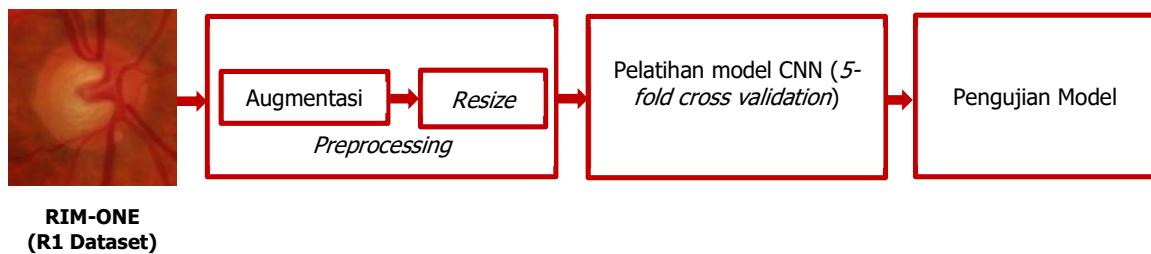
Berbeda dengan penelitian-penelitian sebelumnya, (**Sreng, dkk, 2020**) mengusulkan model deep learning dengan arsitektur *MobileNet*. Penelitian tersebut memperoleh performansi akurasi sebesar 99,7%. (**Mahum, dkk, 2022**) mengusulkan metode ekstraksi fitur citra fundus berbasis *deep learning*, menggunakan CNN. Selanjutnya, fitur ciri yang telah diperoleh diklasifikasikan menggunakan beberapa model klasifikasi, diantaranya, *support vector machine* (SVM), *random forest* (RF), dan *k-nearest neighbor* (KNN) untuk mengklasifikasikan citra fundus kedalam kondisi normal atau glaukoma. Model RF memberikan performansi akurasi terbaik sebesar 98,8% . Demikian pula, (**Ajitha, dkk, 2021**) mengidentifikasi glaukoma berbasis citra fundus dengan membandingkan metode CNN dan SVM. Model CNN memberikan performansi akurasi sebesar 93,86%, sedangkan model SVM mencapai akurasi 95,61%.

Selanjutnya, (**Project & Kumar Shukla, 2020**) membandingkan enam model deep learning yaitu VGG16, VGG19, InceptionV3, InceptionResNetV2, ResNet50 dan DenseNet169 untuk identifikasi penyakit glaukoma. Hasil penelitian tersebut menghasilkan akurasi tinggi 85,19% menggunakan arsitektur DenseNet169. Penelitian terkait sebelumnya oleh (**Fu'adah, dkk, 2021**) mengusulkan model CNN yang terdiri dari tiga *convolutional layer* yang diikuti oleh *pooling layer* serta *dropout layer* pada tahap ekstraksi ciri dengan *a fully connected layer* pada tahap klasifikasi. Performansi akurasi yang dicapai pada penelitian tersebut sebesar 91% dalam mengklasifikasikan kondisi normal dan glaukoma.

Berdasarkan penelitian-penelitian terkait tersebut dapat disimpulkan bahwa identifikasi glaukoma secara otomatis dengan pendekatan *machine learning* dan *deep learning* menunjukkan hasil performansi yang menjanjikan. Namun, pengenalan kondisi citra fundus terbatas untuk dua kondisi yaitu normal dan glaukoma. Selain itu, pada penelitian sebelumnya, secara umum belum mengimplementasikan metode *k-fold cross validation* untuk seleksi model

terbaik. Oleh karena itu, pada penelitian ini kami mengusulkan model CNN yang terdiri dari 5 layer konvolusi pada tahap ekstraksi ciri dan 1-*fully connected layer* pada tahap klasifikasi untuk mengklasifikasikan tahapan penyakit glaukoma yang terdiri dari early, moderate, dan deep. Selain itu sistem juga dapat membedakan kondisi glaukoma dengan kondisi normal dan kondisi OHT yang memiliki karakteristik yang mirip seperti glaukoma. Selanjutnya, untuk mendapatkan performansi model yang optimal dengan kemampuan generalisasi model yang baik dalam mengenali data uji (*unseen data*) kami melakukan optimalisasi model CNN dan mengimplementasikan metode *5-fold cross validation* pada tahap pelatihan model. Hasil performansi model CNN usulan untuk klasifikasi glaukoma selanjutnya akan dibandingkan dengan model arsitektur AlexNet. Arsitektur AlexNet terdiri dari 8 layer, 5 layer konvolusi dan 3-fully connected layer. AlexNet dapat menyelesaikan keterbatasan dalam mengolah jumlah data yang besar dan memiliki resolusi tinggi serta mampu menurunkan tingkat kesalahan hingga 0.5%. Dengan model CNN yang lebih sederhana dari AlexNet, yaitu hanya menggunakan 1- fully connected layer pada tahap klasifikasi, optimalisasi model CNN akan dilakukan untuk memberikan nilai akurasi yang tinggi dan nilai *loss* seminimal mungkin.

2. METODE

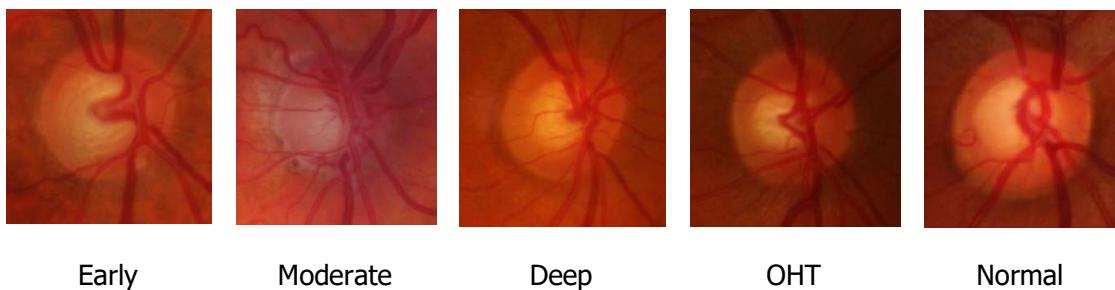


Gambar 1. Blok Diagram Sistem

Gambaran umum dari penelitian ini dapat dilihat pada Gambar 1. Tujuan utama dari penelitian ini adalah optimalisasi *CNN* dan implementasi *5-fold cross validation* untuk meningkatkan performansi sistem dalam mengklasifikasikan citra fundus menjadi lima kategori, antara lain, *early glaucoma*, *moderate glaucoma*, *deep glaucoma*, *OHT*, dan normal. Pada tahap awal citra fundus akan mengalami proses *preprocessing* yang terdiri dari augmentasi untuk memperbanyak data dan *Resize* untuk menyeragamkan data. Proses ekstraksi ciri dan klasifikasi akan dilakukan secara otomatis oleh model *CNN* yang diusulkan. Proses optimalisasi model *CNN* dan implementasi *5-fold cross validation* dilakukan pada tahap pelatihan model menggunakan data latih dan data validasi untuk mendapatkan model terbaik. Selanjutnya, model terbaik yang diperoleh pada tahap pelatihan akan digunakan pada tahap pengujian menggunakan data uji. Pada penelitian ini, dua model akan dibandingkan dalam mengklasifikasikan citra fundus, yaitu menggunakan arsitektur AlexNet dan model *CNN* usulan yang terdiri dari 5 *convolutional layer* pada tahap ekstraksi ciri dan 1 *hidden layer* pada tahap klasifikasi.

2.1 Dataset

Penelitian ini menggunakan citra fundus dari *RIM-ONE* (R1) dataset yang terdiri dari 169 citra fundus yang terbagi dalam lima kelas dengan jumlah data yang berbeda-beda, yaitu 12 citra kondisi *early*, 14 citra kondisi *moderate*, 14 citra kondisi *deep*, 11 citra kondisi OHT dan 118 citra kondisi normal, seperti yang dapat dilihat pada Gambar 2. Untuk mengatasi jumlah dataset yang tidak seimbang, dilakukan proses augmentasi yang terdiri dari *flip horizontal*, dan rotasi secara acak. Setelah tahap augmentasi setiap kelas terdiri dari 1000 citra fundus.

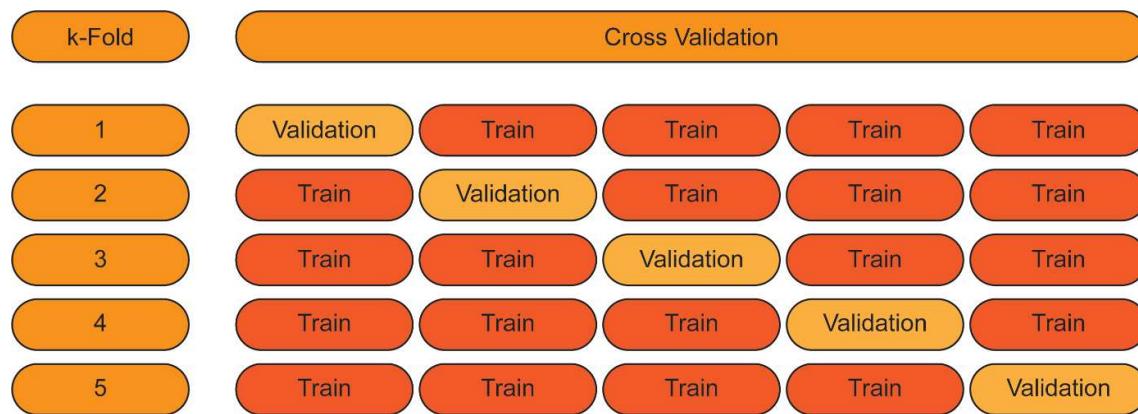


Gambar 2. Citra Fundus RIM-ONE (R1) Dataset

Selanjutnya, dari total 5000 citra fundus, dataset dibagi menjadi 3200 data latih, 800 data validasi dan 1000 data uji untuk memasuki tahap pelatihan model dengan mengimplementasikan metode *5-fold cross validation*.

2.2 Pelatihan model dengan *K-Fold Cross Validation*

K-fold cross validation merupakan salah satu metode statistik yang diimplementasikan untuk mengevaluasi performansi dari model atau algoritma yang telah dirancang. Pada tahap pelatihan dataset dibagi menjadi data latih dan data validasi. Model akan dilatih menggunakan data latih dan divalidasi menggunakan data validasi sebanyak *k-fold* kali. Pada penelitian ini digunakan *5-fold cross validation* dalam mengevaluasi kinerja model. *Five-fold cross validation* merupakan metode yang paling direkomendasikan untuk estimasi kemampuan model dalam generalisasi data. Pada *5-fold cross validation* data dibagi menjadi *5-fold* dengan ukuran sama, dimana *4-fold* akan digunakan sebagai data latih dan *1-fold* digunakan sebagai data validasi. Dari total 5000 data citra fundus, 4000 data akan digunakan pada tahap pelatihan dan 1000 data pada tahap pengujian sebagai data uji. Sejumlah 4000 data yang digunakan pada tahap pelatihan akan dibagi menjadi 5 (setiap *fold* terdiri dari 800 data), sehingga jumlah data latih yang digunakan sebesar 3200 data dan data validasi yang digunakan sebesar 800 data. Selanjutnya proses pelatihan model akan berlangsung sebanyak 5 kali dengan subset data latih dan data validasi yang berbeda pada setiap iterasi pada proses pelatihan seperti yang diilustrasikan pada Gambar 3.

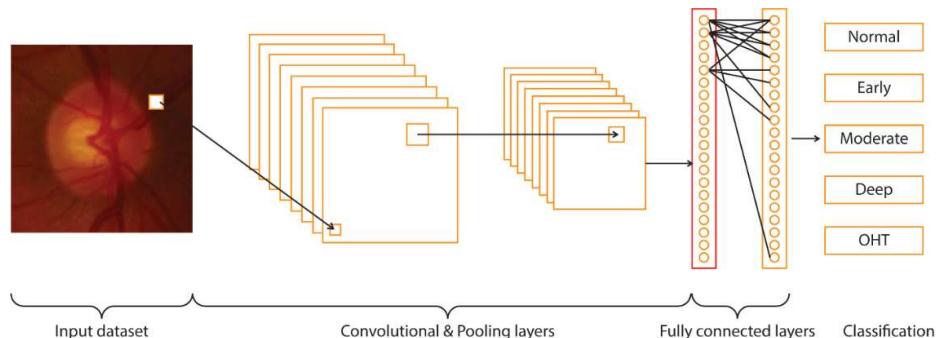


Gambar 3. Ilustrasi Proses *5-fold Cross Validation*

2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan metode *deep learning* yang digunakan untuk mendekripsi, mengidentifikasi, dan mengklasifikasikan data citra. Model CNN banyak digunakan pada pengolahan citra karena tingginya tingkat akurasi yang dapat dicapai. Secara garis besar,

CNN terdiri dari dua bagian utama, yaitu *feature extraction* (ekstrasi ciri citra) dan *classification* (pengklasifikasian) seperti yang ditunjukkan pada Gambar 4 (**Kokila, dkk, 2021**) (**IBM Cloud Education, 2020**).



Gambar 4. Arsitektur Model CNN

Convolutional layer merupakan lapisan pertama pada struktur CNN yang merupakan inti dari CNN dan tempat terjadinya sebagian besar proses komputasi. Pada lapisan ini dilakukan proses konvolusi dengan menggeser matriks filter ke seluruh permukaan citra. Hasil dari proses konvolusi ini adalah *feature map* yang merupakan matriks berisi informasi dari citra yang telah dikonvolusi. *Convolutional layer* dapat diikuti oleh lapisan *convolutional/tambahan* ataupun *pooling layer*. Di setiap convolutional layer akan diikuti oleh aktivasi RELU untuk memilih neuron yang akan aktif (**Suta, dkk, 2019**).

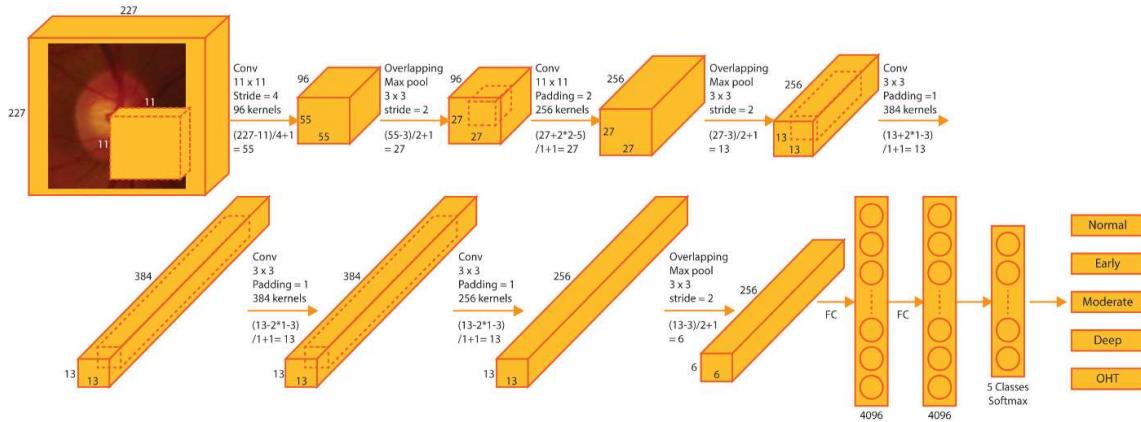
Pooling layer atau yang banyak dikenal dengan lapisan *downsampling* merupakan lapisan untuk mengurangi dimensi dan *feature map* dari lapisan sebelumnya, membantu mengurangi kompleksitas, meningkatkan efisiensi, dan membatasi risiko terjadinya *overfitting*. *Pooling layer* terdiri dari *max pooling* yang mengambil nilai paling tinggi dari data citra, dan *average pooling* yang mengambil nilai rata-rata dari data citra (**Suta, dkk, 2019**) (**IBM Cloud Education, 2020**).

Fully connected layer merupakan lapisan terakhir dalam struktur CNN. Lapisan ini berfungsi untuk melakukan klasifikasi pada citra sesuai dengan informasi yang telah diperoleh pada lapisan sebelumnya. Pada lapisan *fully connected* ini akan dilakukan transformasi pada dimensi data yang sebelumnya multidimensional menjadi linear dengan melakukan proses *flatten* atau *reshape* sehingga semua neuron pada lapisan sebelumnya akan terhubung langsung dan menjadi input pada lapisan ini. Selanjutnya pada tahap klasifikasi multiclass digunakan fungsi aktivasi *softmax* yang akan menghasilkan nilai probabilitas dari 0 hingga 1, dimana penentuan kelas berdasarkan nilai probabilitas paling tinggi (**Rizal, dkk, 2020**).

2.4 Arsitektur AlexNet

Dalam penelitian ini akan diusulkan dua model arsitektur berbasis CNN untuk mengklasifikasikan kondisi Glaukoma, yaitu arsitektur AlexNet dan model CNN sederhana yang terdiri dari 5 convolutional layer. AlexNet merupakan sebuah model berbasis CNN yang dikembangkan oleh Alex Krizhevsky. Arsitektur ini terdiri dari 8 lapisan, yaitu 5 lapisan konvolusi pada tahap ekstraksi ciri dan 3 lapisan fully connected pada tahap klasifikasi seperti yang ditunjukkan pada Gambar 5 (**Fu'adah, dkk, 2021**).

Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma

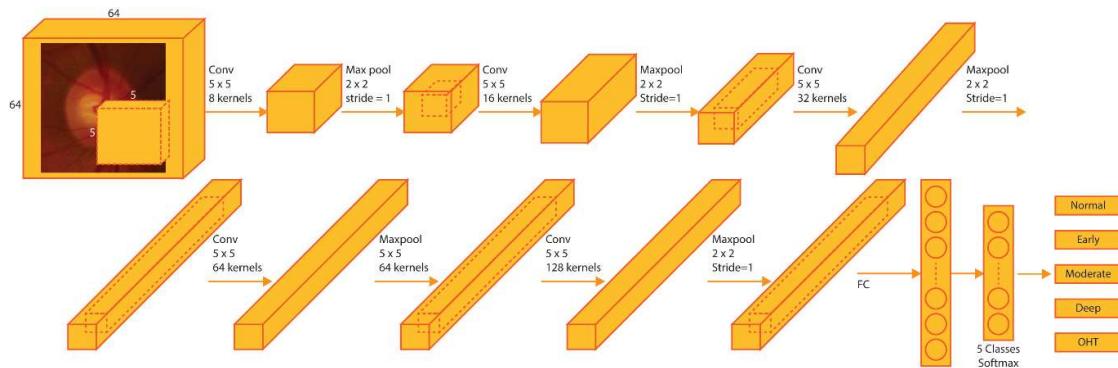


Gambar 5. Arsitektur AlexNet

Bagian *feature extraction* pada AlexNet terjadi pada 5 lapisan konvolusi di awal yang terdiri dari proses konvolusi, *batch normalization*, dan *maxpooling*. Input pada arsitektur ini adalah citra dengan matriks berukuran $224 \times 224 \times 3$. Selanjutnya, matriks input akan dikalikan dengan 96 kernel berukuran $11 \times 11 \times 3$ yang akan digeser sejauh 4 langkah (stride = 4). Kemudian, output akan difilter dengan 256 kernel berukuran $5 \times 5 \times 48$ pada lapisan konvolusi kedua. Output dari convolutional layer kedua akan dihubungkan ke convolutional layer ketiga dengan 384 kernel berukuran $3 \times 3 \times 256$. Convolutional layer keempat dan kelima memiliki 384 kernel berukuran $3 \times 3 \times 192$ dan 256 kernel berukuran $3 \times 3 \times 192$. Selanjutnya fungsi Rel-U akan diaktifasi untuk mengganti nilai negatif menjadi 0. Kemudian proses klasifikasi dilakukan pada 3 lapisan fully connected di akhir arsitektur ini yang masing-masing memiliki 4096 neuron (**Pratiwi, dkk, 2021**).

2.5 Arsitektur Model CNN usulan

Model berikutnya yang diusulkan pada penelitian ini adalah model CNN sederhana yang terdiri dari 5 *convolutional layer* pada tahap ekstraksi ciri. Setiap layer konvolusi akan diikuti oleh *maxpooling* 2D dan aktivasi fungsi REL-U. Input citra akan difilter pada *convolutional layer* pertama sampai kelima dengan jumlah kernel berturut-turut adalah 8, 16, 32, 64 dan 128 dengan ukuran filter 5×5 pada setiap layernya. Lapisan fully connected sebagai lapisan terakhir menggunakan satu *hidden layer* dengan 10 *neuron* dan fungsi aktivasi *Softmax* untuk klasifikasi 5 kondisi citra fundus. Arsitektur dari model CNN usulan ditunjukkan oleh Gambar 6.



Gambar 6. Arsitektur Model CNN Usulan

2.6 Evaluasi Performansi Sistem

Kinerja sistem dievaluasi dengan menggunakan *Confusion Matrix* untuk mendapatkan nilai akurasi, presisi, recall, dan F1-score. Persamaan (1), Persamaan (2), Persamaan (3), dan Persamaan (4) digunakan untuk mengukur efektivitas sistem dalam mengklasifikasikan kondisi citra fundus *early*, *moderate*, *deep*, *OHT*, dan normal.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$\text{Precision} = TP/(TP + FP) \quad (2)$$

$$\text{Recall} = TP/(TP + FN) \quad (3)$$

$$\text{F1 score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

Pada Persamaan (1), Persamaan (2), dan Persamaan (3), *True Positive (TP)* merupakan kondisi saat model berhasil mengklasifikasikan kelas positif dengan akurat sebagai kelas positif. Sebaliknya, *True Negative (TN)*, merupakan kondisi saat model berhasil mengklasifikasikan kelas negatif dengan akurat sebagai kelas negatif. *False Positive (FP)* merupakan kondisi saat model keliru mengklasifikasikan kelas negative sebagai sebagai kelas positif. Sedangkan *False Negative (FN)* merupakan kondisi saat model keliru mengklasifikasikan kelas positif sebagai kelas negatif (**Novakovic, dkk, 2017**)

3. HASIL DAN DISKUSI

Tahap pelatihan dilakukan dengan mengimplementasikan *5-fold cross validation* menggunakan data latih yang terdiri dari 3200 citra fundus dan data validasi yang terdiri dari 800 citra fundus. Selanjutnya model terbaik digunakan pada tahap pengujian menggunakan data uji yang terdiri dari 1000 citra fundus. Tahap pelatihan dan pengujian dilakukan dengan beberapa skenario untuk optimalisasi model CNN dan mengetahui pengaruh dari *hyperparameters* yang memberikan performansi terbaik. Skenario pertama dilakukan pengujian untuk mendapatkan jumlah *epoch* optimal bagi model dalam melakukan tahap pelatihan sehingga memperoleh performansi terbaik. *Epoch* merupakan satu putaran ketika keseluruhan dataset telah melewati tahap pelatihan pada *Neural Network*. Penelitian ini menguji beberapa nilai *epoch*, antara lain, 10, 25, 50, dan 100. Pada saat pengujian *epoch* ini ukuran *batch size* yang digunakan sebesar 64, *learning rate* 0.001, dan menggunakan *optimizer* Adam.

Tabel 1. Perbandingan performansi akurasi dan loss dalam beberapa nilai epoch

Arsitektur	Epoch	Akurasi Latih	Akurasi Validasi	Akurasi Uji	Loss
AlexNet	10	86.11%	86.89%	85.41%	0.4243
	25	98.79%	89.38%	90.29%	0.3772
	50	99.94%	99.75%	99.80%	0.0029
	100	100%	100%	100%	0.00002
Model CNN Usulan	10	97.98%	99.12%	98.68%	0.0579
	25	100%	99.62%	100%	0.0007
	50	100%	99.62%	99.80%	0.0102
	100	100%	99.75%	100%	0.0002

Berdasarkan Tabel 1, dapat dilihat bahwa seiring bertambahnya nilai *epoch* baik menggunakan arsitektur AlexNet maupun model CNN yang diusulkan nilai akurasi meningkat diiringi dengan nilai *loss* yang menurun. Hal ini dikarenakan nilai *epoch* yang kecil menyebabkan kegagalan proses *update* bobot mencapai titik optimalnya. Pada saat nilai *epoch* 10 model AlexNet memberikan performansi akurasi yang lebih rendah dan nilai *loss* yang lebih besar

dibandingkan dengan model CNN usulan yang memberikan performansi akurasi yang cukup baik meskipun masih menggunakan nilai *epoch* yang masih minimum. Performansi terbaik kedua model diperoleh pada saat nilai epoch 100 dengan memberikan akurasi data latih dan data validasi yang tidak *overfitting* dan tidak *underfitting*. Hal ini ditunjukkan dengan nilai performasi akurasi data latih dan data validasi yang tidak jauh berbeda. Selain itu, model memberikan performansi akurasi data uji 100%. Berdasarkan hal tersebut, dapat disimpulkan bahwa nilai *epoch* 100 merupakan nilai yang optimal untuk digunakan, baik untuk model AlexNet dan model CNN usulan.

Selanjutnya, pada skenario kedua dilakukan pengujian pengaruh *batch size* terhadap performansi sistem. Pada saat pengujian batch size kami menggunakan parameter epoch 100, *optimizer Adam* dan *learning rate* 0.001. Nilai *batch size* yang digunakan antara lain, 16, 32, dan 64. Hasil pengujian batch size untuk model AlexNet dan model CNN usulan dapat dilihat pada Tabel 2.

Tabel 2. Perbandingan performansi akurasi dan loss dalam beberapa nilai *batch size*

Arsitektur	Batch Size	Akurasi Latih	Akurasi Validasi	Akurasi Uji	Loss
AlexNet	16	99.97%	100%	100%	0.0003
	32	99.94%	99.87%	100%	0.0008
	64	100%	100%	100%	0.00002
Model CNN usulan	16	100%	99.75%	99. 59%	0.0153
	32	99.22%	99.37%	99%	0.0283
	64	100%	99.75%	100%	0.0002

Ukuran *batch size* menunjukkan berapa banyak jumlah sampel yang akan diambil dari dataset pelatihan. Berdasarkan Tabel 2 model AlexNet memberikan performansi akurasi untuk data latih, data validasi dan data uji sebesar 100% dengan nilai loss yang rendah untuk semua nilai *batch size*. Hal serupa ditunjukkan oleh model CNN usulan, meskipun tidak memberikan performansi 100% pada saat penggunaan nilai *batch size* 16 dan 32. Performansi terbaik kedua model diperoleh pada saat nilai batch size 64 yang ditunjukkan dengan nilai akurasi yang maksimal dan *loss* yang minimal.

Pada skenario ketiga, dilakukan pengujian beberapa nilai *learning rate*, antara lain, *learning rate* dengan nilai 0.1, 0.01, 0.001, dan 0.0001. Parameter yang digunakan pada saat pengujian *learning rate* antara lain jumlah *epoch* 100, *batch size* 64, dengan *optimizer Adam*. Hasil pengujian *learning rate* untuk model AlexNet dan model CNN usulan dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan performansi akurasi dan loss dalam beberapa nilai *learning rate*

Arsitektur	Learning Rate	Akurasi Latih	Akurasi Validasi	Akurasi Uji	Loss
AlexNet	0.0001	100%	100%	100%	0.0001
	0.001	100%	100%	100%	0.00002
	0.01	70.44%	70.25%	75%	0.65
Model CNN usulan	0.1	18.38%	20.62%	21.79%	1.609
	0.0001	99.69%	99.12%	99. 40%	0.0194
	0.001	100%	99.75%	100%	0.0002
	0.01	20.66%	18.25%	19.79%	0.61
	0.1	19.31%	20.87%	19.40%	1.609

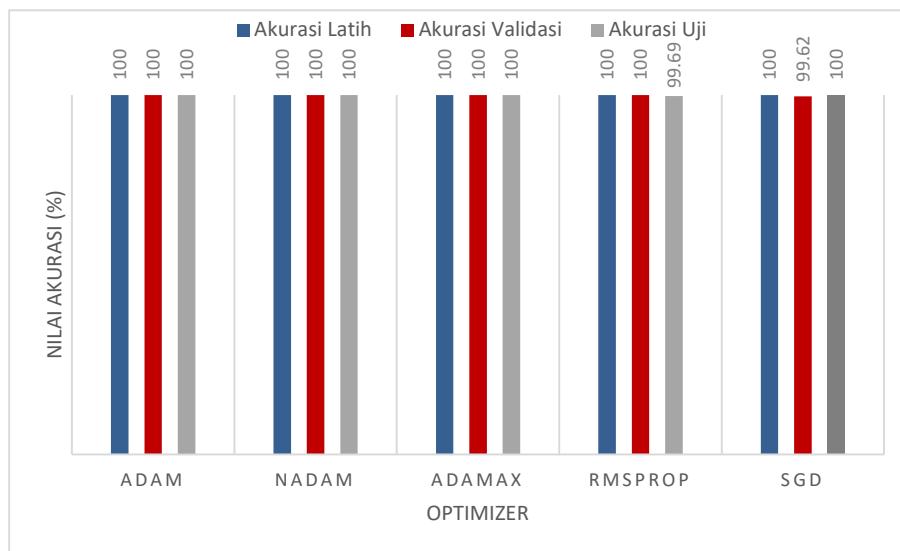
Learning rate merupakan parameter pada tahap pelatihan yang digunakan untuk perhitungan nilai koreksi bobot pada saat proses pelatihan berlangsung. Berdasarkan Tabel 3, dapat dilihat

bawa pada saat nilai learning rate besar, performansi akurasi untuk data latih, data validasi, dan data uji baik yang dihasilkan model AlexNet maupun model CNN usulan bernali sangat rendah dan memberikan performansi terbaik saat *learning rate* bernilai 0.001. Hal ini dikarenakan nilai *learning rate* yang semakin besar mengakibatkan proses pelatihan yang semakin cepat, akan tetapi berpengaruh terhadap berkurangnya tingkat ketelitian *Neural Network*. Sedangkan, semakin kecil nilai *learning rate*, tingkat ketelitian *Neural Network* semakin besar, akan tetapi akan memerlukan proses pelatihan yang lebih lama.

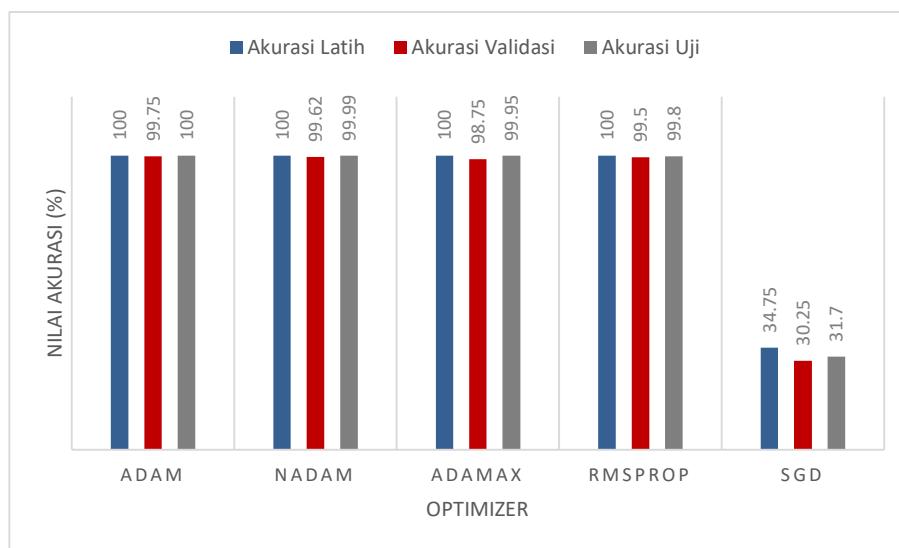
Skenario pengujian keempat adalah pengujian jenis *optimizer* yang terdiri dari *optimizer Adam, Nadam, Adamax, RMSprop, dan SGD*. Hasil pengujian jenis *optimizer* yang digunakan untuk arsitektur AlexNet dan model CNN yang diusulkan dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan performansi akurasi dan *loss* dalam beberapa nilai *optimizer*

Arsitektur	Optimizer	Akurasi Latih	Akurasi Validasi	Akurasi Uji	Loss
AlexNet	Adam	100%	100%	100%	0.00002
	Nadam	100%	100%	100%	0.0004
	Adamax	100%	100%	100%	0.00003
	RMSprop	100%	100%	99.69%	0.0001
	SGD	100%	99.62%	100%	0.0105
Model CNN usulan	Adam	100%	99.75%	100%	0.0002
	Nadam	100%	99.62%	99.99%	0.004
	Adamax	100%	98.75%	99.95%	0.0082
	RMSprop	100%	99.50%	99.80%	0.0119
	SGD	34.75%	30.25%	31.70%	1.522



Gambar 7. Grafik perbandingan nilai akurasi untuk berbagai *optimizer* pada Arsitektur AlexNet



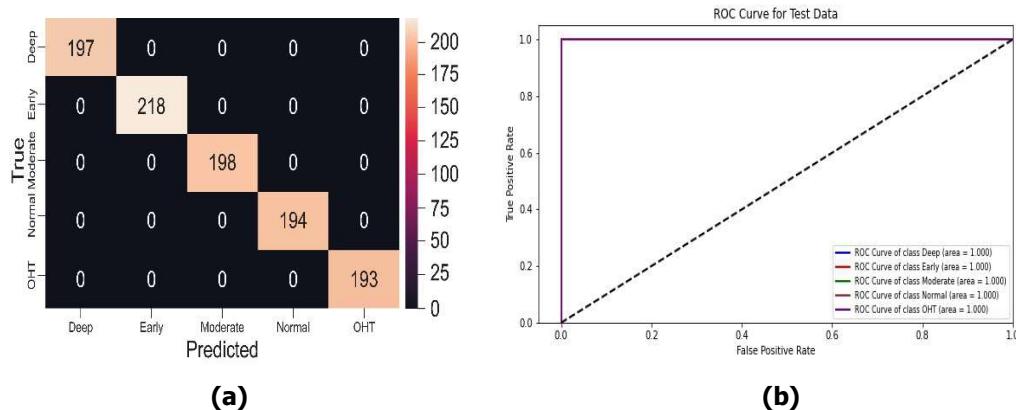
Gambar 8. Grafik perbandingan nilai akurasi untuk berbagai *optimizer* pada model CNN usulan

Penggunaan *optimizer* bertujuan untuk mendapatkan performansi sistem yang optimal dan meminimalisir nilai *error*. Berdasarkan Tabel 4, dapat dilihat bahwa secara umum penggunaan *optimizer Adam*, *Nadam*, *Adamax* dan *RMSprop* memberikan performansi akurasi yang tinggi dan *loss* yang minimum baik menggunakan arsitektur AlexNet maupun menggunakan model CNN usulan. Sedangkan untuk *optimizer SGD* memberikan performansi akurasi 100 % untuk arsitektur AlexNet seperti yang ditunjukan pada Gambar 7, dan hanya mendapatkan akurasi sebesar 31.70 % untuk model CNN usulan seperti yang ditunjukan pada Gambar 8. Selain itu nilai *loss* yang dihasilkan *optimizer SGD* pada model CNN usulan menghasilkan nilai *loss* yang sangat tinggi sebesar 1.522. Selain berdasarkan nilai akurasi, tingkat keberhasilan model dapat dianalisis juga berdasarkan nilai *loss*. Proses optimalisasi dilakukan dengan cara meng-*update* nilai vektor bobot menggunakan algoritma *optimizer* tertentu dengan tujuan untuk meminimalisir nilai *loss*. Nilai *loss* tinggi yang dihasilkan *optimizer SGD* pada model CNN usulan menunjukkan tingginya tingkat kesalahan model dalam memprediksi data secara tepat. Oleh karena itu, nilai akurasi yang diperoleh pada saat menggunakan *optimizer SGD* sangat rendah dibandingkan dengan algoritma *optimizer* lainnya. Perbedaan nilai akurasi yang sangat signifikan pada model CNN usulan seperti yang ditunjukan Gambar 8, nilai akurasi uji 31.7% menggunakan *optimizer SGD* dan nilai akurasi uji 100% menggunakan *optimizer Adam* menunjukan pentingnya pemilihan algoritma *optimizer* yang dapat memberikan performansi sistem yang optimal.

Tabel 5. Nilai Parameter Performansi Sistem untuk Setiap Kelas

Arsitektur	Kelas	Presisi	Recall	<i>F1-score</i>
AlexNet	Early	1.0	1.0	1.0
	Moderate	1.0	1.0	1.0
	Deep	1.0	1.0	1.0
	OHT	1.0	1.0	1.0
	Normal	1.0	1.0	1.0
Model CNN usulan	Early	1.0	1.0	1.0
	Moderate	1.0	1.0	1.0
	Deep	1.0	1.0	1.0
	OHT	1.0	1.0	1.0
	Normal	1.0	1.0	1.0

Berdasarkan Tabel 5. Dapat dilihat bahwa performansi terbaik sistem baik menggunakan model AlexNet, maupun model CNN usulan memberikan nilai presisi, *recall*, dan *F1-score* sebesar 1. Hal ini menunjukkan bahwa model berhasil mengklasifikasikan setiap kelas citra fundus sesuai dengan kelasnya termasuk kategori *early*, *moderate*, *Deep*, *OHT* dan normal sesuai kelasnya seperti yang ditunjukkan oleh *Confusion Matrix* pada Gambar 9a dan AUC score bernilai 1 untuk setiap kelasnya seperti yang ditunjukkan pada Gambar 9b.

**Gambar 9. Performansi Sistem. (a) Confusion Matrix; (b) ROC curve untuk data uji**

Performansi terbaik ini diperoleh pada saat penggunaan *epoch* 100, *batch size* 64, *learning rate* 0.001, dan *optimizer Adam*. Nilai parameter terbaik tersebut diperoleh dari serangkaian pengujian *hyperparameter* untuk mendapatkan model yang paling optimal. Epoch merupakan hyperparameter yang menunjukkan berapa kali *gradient descent* atau algoritma pengoptimalan akan dilakukan pada tahap pelatihan untuk mendapatkan *loss* paling minimum dan akurasi yang tinggi. Berdasarkan hasil pengujian, semakin kecil nilai *epoch*, nilai akurasi yang diperoleh semakin kecil. Hal ini dikarenakan nilai *epoch* yang kecil tidak cukup dalam mengupdate bobot untuk mencapai titik optimalnya. Sedangkan, nilai *epoch* yang terlalu besar juga dapat menyebabkan *overfitting*. Oleh karena itu diperlukan serangkaian pengujian dengan menguji beberapa nilai *epoch* dan menganalisis hasil akurasi latih dan akurasi validasi yang tidak menyebabkan *overfitting*. Pada penelitian ini nilai *epoch* 100 memberikan performansi terbaik dengan menghasilkan nilai akurasi latih, validasi dan uji yang tidak *overfitting*. Selanjutnya, penentuan nilai *batch size* sangat berpengaruh dalam mengontrol keakuratan estimasi nilai *error gradient* pada tahap pelatihan. Pada penelitian ini, performansi nilai akurasi *batch size* yang diuji tidak terlalu berbeda secara signifikan. Akan tetapi, nilai *batch size* 64 memberikan performansi *loss* yang paling minimum. Sehingga, *batch size* 64 merupakan nilai *batch size* yang paling optimal untuk digunakan pada saat pelatihan model

CNN pada penelitian ini. *Hyperparameter* lainnya yang mempengaruhi performansi sistem adalah *learning rate*. Semakin besar nilai learning rate waktu komputasi akan semakin cepat, akan tetapi ketelitian jaringan akan berkurang. Sebaliknya, semakin kecil nilai learning rate, tingkat ketelitian akan semakin tinggi akan tetapi membutuhkan waktu training yang lebih lama. Oleh karena itu diperlukan penentuan nilai *learning rate* yang paling optimal dalam memberikan performansi akurasi sistem yang tinggi dengan waktu komputasi yang tidak terlalu lama. Berdasarkan hasil pengujian nilai learning rate 0.1-0.0001, nilai learning rate 0.001 merupakan nilai *learning rate* yang paling optimal untuk digunakan pada penelitian ini berdasarkan performansi akurasi latih dan validasi yang dihasilkan. Selanjutnya, untuk meningkatkan performansi akurasi dan menghasilkan loss seminimal mungkin diperlukan pemilihan optimizer yang tepat. Penentuan *optimizer* yang digunakan dapat dianalisis berdasarkan nilai akurasi dan nilai loss yang dihasilkan. *Optimizer Adam* memberikan performansi akurasi uji sebesar 100 % baik untuk arsitektur AlexNet dan model CNN usulan serta nilai *loss* 0.00002 untuk arsitektur AlexNet dan nilai loss 0.0002 untuk model CNN usulan yang merupakan nilai *loss* paling minimal dibandingkan menggunakan *optimizer* lainnya.

4. KESIMPULAN

Penelitian ini melakukan optimalisasi model CNN dengan melakukan serangkaian pengujian dalam memilih *hyperparameter* yang terdiri dari pengujian *epoch*, *batch size*, *learning rate*, dan *optimizer* yang bertujuan untuk memperoleh parameter optimal yang memberikan performansi sistem terbaik. Implementasi *5-fold cross validation* menggunakan 3200 data latih dan 800 data validasi dilakukan pada tahap pelatihan model untuk estimasi kemampuan generalisasi model terhadap data uji. Selanjutnya model akan digunakan untuk klasifikasi data uji yang terdiri dari 1000 citra fundus. Penelitian ini mempertimbangkan dua model pelatihan *deep learning* yaitu menggunakan arsitektur AlexNet dan model CNN dengan arsitektur usulan yang terdiri dari 5 *convolutional layer* pada tahap ekstraksi ciri dan satu hidden layer pada tahap klasifikasi. Berdasarkan hasil pengujian, penentuan *hyperparameter* sangat berpengaruh terhadap performansi sistem. Performansi terbaik diperoleh baik menggunakan arsitektur AlexNet maupun model CNN yang diusulkan dengan memperoleh nilai akurasi data uji sebesar 100 % dan nilai presisi, recall, *F1-score*, serta *AUC score* bernilai 1. Hal ini menunjukkan bahwa model mampu mengklasifikasikan data uji sesuai dengan kelasnya secara akurat. Sebagai rencana riset selanjutnya, untuk menguji kemampuan generalisasi dari model dapat dilakukan dengan penggunaan dataset yang berbeda dengan jumlah data yang lebih besar.

DAFTAR RUJUKAN

- Ajitha, S., Akkara, J. D., & Judy, M. V. (2021). Identification of glaucoma from fundus images using deep learning techniques. *Indian Journal of Ophthalmology*, 69(10), 2702–2709.
https://doi.org/10.4103/ijo.IJO_92_21
- An, G., Omodaka, K., Hashimoto, K., Tsuda, S., Shiga, Y., Takada, N., Kikawa, T., Yokota, H., Akiba, M., & Nakazawa, T. (2019). Glaucoma Diagnosis with Machine Learning Based on Optical Coherence Tomography and Color Fundus Images. *Journal of Healthcare Engineering*, 2019, 4061313. <https://doi.org/10.1155/2019/4061313>
- Fu'adah, Y. N., Sa'idah, S., Wijayanto, I., Ibrahim, N., Rizal, S., & Magdalena, R. (2021).

- Computer Aided Diagnosis for Early Detection of Glaucoma Using Convolutional Neural Network (CNN). In Triwiyanto, H. A. Nugroho, A. Rizal, & W. Caesarendra (Eds.), *Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics*, (pp. 467–475). Springer Singapore.
- Fu'adah, Y. N., Wijayanto, I., Pratiwi, N. K. C., Taliningsih, F. F., Rizal, S., & Pramudito, M. A. (2021). Automated Classification of Alzheimer's Disease Based on {MRI} Image Processing using Convolutional Neural Network ({CNN}) with {AlexNet} Architecture. *Journal of Physics: Conference Series*, 1844(1), 12020. <https://doi.org/10.1088/1742-6596/1844/1/012020>
- IBM Cloud Education. (2020). *Convolutional Neural Networks*. IBM Cloud.
- Kokila, B., Devadharshini, M. S., Anitha, A., & Abisheak Sankar, S. (2021). Brain Tumor Detection and Classification Using Deep Learning Techniques based on MRI Images. *Journal of Physics: Conference Series*, 1916(1). <https://doi.org/10.1088/1742-6596/1916/1/012226>
- Mahum, R., Rehman, S. U., Okon, O. D., Alabrah, A., Meraj, T., & Rauf, H. T. (2022). A Novel Hybrid Approach Based on Deep CNN to Detect Glaucoma Using Fundus Imaging. *Electronics*, 11(1). <https://doi.org/10.3390/electronics11010026>
- Novakovic, J., Veljovi, A., Ilic, S., Papic, Z., & Tomovic, M. (2017). Evaluation of Classification Models in Machine Learning. *Theory and Applications of Mathematics & Computer Science*, 7(1), 39–46. <https://uav.ro/applications/se/journal/index.php/TAMCS/article/view/158>
- Pratiwi, N. K. C., Fu'adah, Y. N., & Edwar, E. (2021). Early Detection of Deforestation through Satellite Land Geospatial Images based on CNN Architecture. *Jurnal Infotel*, 13(2), 54–62. <https://doi.org/10.20895/infotel.v13i2.642>
- Project, R., & Kumar Shukla, M. (2020). *Classification of Different Stages of Glaucoma Using Deep Learning Approaches*.
- Rizal, Syamsul., Ibrahim, Nur., Pratiwi, Nor., Saidah, Sofia., & Fu'adah. (2020). Deep Learning untuk Klasifikasi Diabetic Retinopathy menggunakan Model EfficientNet. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 8. 693. 10.26760/elkomika.v8i3.693.
- Sreng, S., Maneerat, N., Hamamoto, K., & Win, K. Y. (2020). Deep Learning for Optic Disc Segmentation and Glaucoma Diagnosis on Retinal Images. *Applied Sciences*, 10(14). <https://doi.org/10.3390/app10144916>
- Suta, I. B. L. M., Hartati, R. S., & Divayana, Y. (2019). Diagnosa Tumor Otak Berdasarkan

Optimasi *Convolutional Neural Network* dan *K-Fold Cross Validation* pada Sistem Klasifikasi Glaukoma

Citra MRI (Magnetic Resonance Imaging). *Majalah Ilmiah Teknologi Elektro*, 18(2).
<https://doi.org/10.24843/mite.2019.v18i02.p01>

Tham, Y.-C., Li, X., Wong, T. Y., Quigley, H. A., Aung, T., & Cheng, C.-Y. (2014). Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis. *Ophthalmology*, 121(11), 2081–2090.
<https://doi.org/10.1016/j.ophtha.2014.05.013>

Wu, C.-W., Shen, H.-L., Lu, C.-J., Chen, S.-H., & Chen, H.-Y. (2021). Comparison of Different Machine Learning Classifiers for Glaucoma Diagnosis Based on Spectralis OCT. *Diagnostics*, 11(9). <https://doi.org/10.3390/diagnostics11091718>