

Pendeteksian *Retinopathy of Prematurity* (ROP) pada Bayi Prematur Menggunakan Metode Convolutional Neural Network (CNN)

Annisa Zahra, Alexandria Samantha Nicole, Angelica Patricia Djaya Saputra, Arundhati Naysa Ekhaputri, Hosia Josindra Saragih, Latifa Aulia Esmananda

Dibimbing oleh: Devvi Sarwinda, S.Si. M.Kom.

Departemen Matematika, Universitas Indonesia, Depok 16424, Indonesia

Dikumpulkan sebagai laporan akhir kuliah Pemodelan Matematis, Semester Gasal 2022

Abstrak

Retinopathy of prematurity (ROP) adalah kelainan proliferasi progresif pembuluh darah retina bayi prematur yang dapat menyebabkan ablasio retina dan kebutaan. Kelainan proliferasi progresif merupakan pembelahan secara terus menerus dalam kuantitas yang banyak pada pembuluh darah. Faktor risiko penderita yang paling berpengaruh adalah prematuritas dan berat badan lahir rendah. Dalam beberapa tahun terakhir, terdapat upaya yang signifikan untuk mengotomatisasi diagnosis menggunakan *machine learning*. Makalah ini disusun dengan menggabungkan segmentasi objek dan *convolutional neural network* (CNN) untuk mengklasifikasikan apakah seorang bayi prematur mengidap ROP atau tidak berdasarkan *image fundus* mata. Data yang digunakan didapatkan melalui laman kaggle yaitu <https://www.kaggle.com/code/solennollivier/rop-2classclassification/data> berupa 91 citra fundus mata yang tidak ROP, penderita ROP Stage 1, penderita ROP Stage 2, dan penderita ROP Stage 3 dengan format jpg. Penelitian ini dilakukan dengan membuat 2 model yaitu model CNN menggunakan VGG-19 tanpa augmentasi data dan model CNN menggunakan VGG-19 dengan augmentasi data. Didapatkan hasil model terbaik untuk melakukan pendeteksian ROP (*Retinopathy of Prematurity*) pada bayi prematur adalah model CNN menggunakan VGG-19 dengan augmentasi data, dengan 96% accuracy, 96% precision, 96% recall, 96% F1-Score, dan 0.14 Loss.

Kata kunci: *Retinopathy of prematurity, deep learning, convolutional neural network.*

1. Pendahuluan

1.1. Latar Belakang

Bayi prematur adalah bayi yang lahir sebelum usia kandungan mencapai 37 minggu. Saat ini di Indonesia angka kelahiran bayi prematur mencapai angka golongan tinggi. Menurut Kompas (2015), pada tahun 2010 Indonesia menempati posisi ke-5 tertinggi di dunia dengan angka kelahiran prematur sekitar 675.000 bayi. Sedangkan untuk data dunia, Gracia (2019) menyatakan data yang diperoleh di Amerika Serikat dimana dari 28.000 bayi yang lahir dengan kondisi prematur terdapat 16.000 diantaranya mengalami penyakit mata jenis *Retinopathy of Prematurity* [1].

Usia kehamilan kurang dari 31 minggu pada bayi prematur memiliki risiko lebih tinggi untuk mengalami penyakit mata dimana berpotensi merusak penglihatan. Gangguan fungsi penglihatan yang sering dialami adalah gangguan pertumbuhan saraf mata atau retina yaitu *Retinopathy of Prematurity* (ROP).

Menurut dr. Soejono dalam Solahuddin (2019) menyatakan pada tahun 2019 diperoleh dari sekitar 4,5 juta kelahiran 30% diantaranya lahir prematur dan memiliki risiko mengalami ROP [2].

Penyakit *Retinopathy of prematurity* adalah kelainan proliferasi progresif pembuluh darah retina bayi prematur yang dapat menyebabkan ablasi retina dan kebutaan. Kelainan proliferasi progresif itu sendiri adalah pembelahan secara terus menerus dalam kuantitas yang banyak pada pembuluh darah. Faktor risiko penderita yang paling berpengaruh adalah prematuritas dan berat badan lahir rendah. Oleh karena itu tingginya angka kelahiran prematur dapat menyebabkan risiko potensi terkena penyakit *Retinopathy of prematurity* (ROP) yang dapat menyebabkan kebutaan.

Pada dasarnya, gejala ROP pada bayi prematur terbagi menjadi lima tahapan, yaitu untuk stadium I: pertumbuhan pembuluh darah agak abnormal dan bisa sembuh dengan sendirinya, stadium II: pertumbuhan pembuluh darah cukup abnormal dan masih bisa sembuh sendiri, stadium III: pertumbuhan pembuluh darah sangat abnormal ke arah pusat mata, stadium IV: retina terlepas sebagian, pembuluh darah abnormal menarik retina jauh dari dinding mata, dan stadium V: retina benar-benar terlepas [3,4].

Oleh karena tingginya angka kelahiran bayi prematur berpotensi mengalami gangguan penglihatan yang disebabkan oleh ROP dimana dapat menyebabkan kebutaan pada bayi, maka peneliti ingin melakukan pendeteksian *Retinopathy of prematurity* (ROP) pada bayi prematur.

1.2. Rumusan Masalah

Pemodelan yang kami lakukan harapannya dapat menjawab premis masalah yang dirumuskan sebagai berikut:

1. Bagaimana cara kerja metode *Convolutional Neural Network* dalam mendeteksi *Retinopathy of prematurity* (ROP) pada bayi prematur?
2. Bagaimana analisis kinerja metode *Convolutional Neural Network* dalam mendeteksi *Retinopathy of prematurity* (ROP) pada bayi prematur?
3. Bagaimana pengaruh penggunaan augmentasi data dalam mendeteksi *Retinopathy of prematurity* (ROP) pada bayi prematur?

1.3. Tujuan Penelitian

1. Menjelaskan cara kerja metode *Convolutional Neural Network* dalam mendeteksi *Retinopathy of prematurity* (ROP) pada bayi prematur.

2. Menganalisis kinerja model tanpa augmentasi data dan dengan augmentasi data untuk mendapatkan model terbaik dalam penerapan kasus pendeteksian *Retinopathy of prematurity* (ROP) pada bayi prematur.
3. Mengetahui pengaruh augmentasi data pada model dalam penerapan kasus pendeteksian *Retinopathy of prematurity* (ROP) pada bayi prematur.

1.4. Batasan Masalah

Pemodelan yang kami lakukan dibatasi dengan menggunakan dataset publik (<https://www.kaggle.com/code/solennollivier/rop-2classclassification/data>) yang menyediakan 91 data gambar fundus mata. Data gambar merupakan fundus mata dari pasien yang mengalami ROP pada stage 1-3 dan fundus mata normal atau tanpa mengalami ROP.

Untuk menyederhanakan proses pemodelan, model tidak mempertimbangkan faktor demografi lain yang tidak tersedia pada dataset, seperti jenis kelamin, usia gestasi yang merupakan waktu yang dibutuhkan seorang ibu selama masa konsepsi hingga kelahiran (Spong, 2013), dan usia dari bayi.

2. Tinjauan Pustaka

2.1. Machine Learning

Machine Learning adalah cabang dari AI yang fokus belajar dari data, yaitu fokus pada pengembangan sistem yang mampu belajar secara “mandiri” tanpa harus berulang kali diprogram manusia. Menurut Arthur Samuel (1959), *Machine Learning* adalah bidang studi yang memberikan komputer kemampuan untuk belajar tanpa diprogram secara eksplisit [5].

Machine learning adalah tentang mengekstraksi pengetahuan dari data. Ini adalah bidang penelitian yang merupakan persimpangan dari statistik, *Artificial Intelligence*, dan *Computer Science* dan dikenal sebagai analitik prediktif atau pembelajaran statistik. Aplikasi dari metode *Machine Learning* dalam beberapa tahun terakhir menjadi umum dalam kehidupan sehari-hari, mulai dari rekomendasi film, musik, makanan apa yang harus dipesan, dan lainnya [6].

2.2. Deep Learning

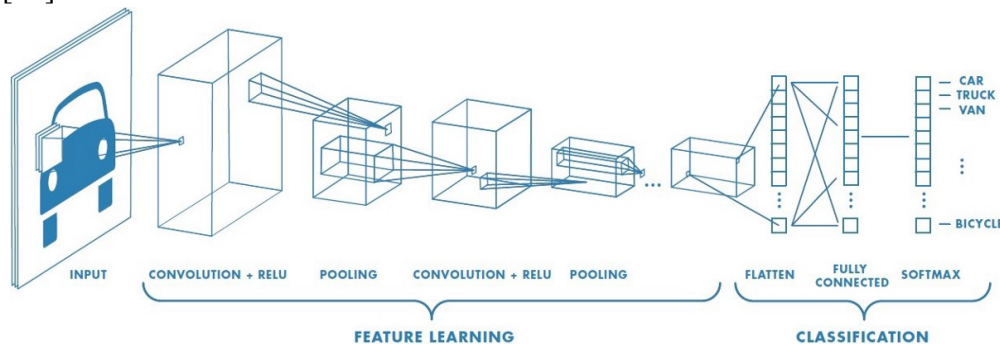
Deep learning adalah bagian dari *machine learning* yang merupakan sebuah implementasi dari *artificial neural networks*. Pada dasarnya, deep learning merupakan jaringan saraf tiruan dengan tiga atau lebih layer (*input layer*, *hidden layer*, dan *output layer*) [7]. *Deep learning* terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. *Deep learning* sangat baik untuk diterapkan pada *supervised learning*, *unsupervised learning* dan *semi-supervised learning* maupun untuk *reinforcement learning* dalam berbagai aplikasi seperti pengenalan citra, suara, klasifikasi teks, dan sebagainya [8].

Deep learning menemukan struktur rumit dalam kumpulan *big data* dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya [9].

2.3. Convolutional Neural Network

Convolutional Neural Network merupakan salah satu jenis metode dari *deep learning* yang sering digunakan untuk mengolah data *image*, yakni jenis jaringan saraf tiruan yang umumnya digunakan dalam pengenalan dan pemrosesan gambar. Algoritma ini dirancang khusus untuk memproses data input berupa data *image* (citra visual), menetapkan kepentingan (bobot dan bias

yang dapat dipelajari) untuk berbagai aspek/objek dalam gambar dan dapat membedakan satu dari yang lain [13].



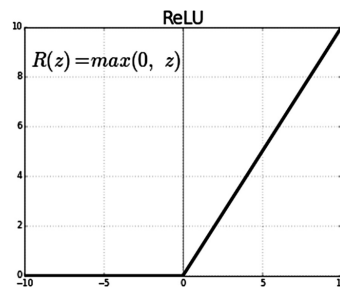
Gambar 2. *Convolutional Neural Network*

Sumber : towardsdatascience.com

Arsitektur pada algoritma CNN terdiri atas 2 komponen, yakni komponen *feature learning* dan komponen klasifikasi.

2.3.1 ReLU

ReLU (Rectified Linear Unit) adalah fungsi aktivasi yang bertujuan untuk memperkenalkan non linearitas di dalam jaringan konvolusi [14]. fungsi aktivasi ReLU ini merupakan fungsi aktivasi yang paling sering digunakan secara default untuk melakukan sebagian besar tugas *deep learning*. Variannya biasanya digunakan untuk tujuan tertentu di mana mereka mungkin memiliki sedikit keunggulan dibandingkan ReLU.



Gambar x. Fungsi aktivasi ReLU [15]

Rumus dari fungsi aktivasi ReLU ini adalah sebagai berikut: $R(z) = \max(0, z)$

2.3.2 Komponen Feature Learning

Feature learning adalah proses *encoding* dari sebuah *image* menjadi *features* yang berupa angka-angka yang merepresentasikan *image* tersebut. Pada bagian ini, terdapat beberapa *layer* yang saling bekerjasama untuk mengambil ciri dari *image*. Berikut adalah penjelasan *layer-layer* pada komponen ini:

2.3.2.1 Convolutional Layer

Convolution dijalankan pada data input menggunakan sebuah filter atau kernel yang kemudian digunakan untuk memetakan fitur. Operasi tersebut dilakukan dengan menggeser filter di atas input. Di setiap lokasi, perkalian matriks dilakukan dan menjumlahkan hasilnya ke dalam peta fitur. Beberapa operasi *convolution* pada data input akan dilakukan menggunakan filter yang berbeda.

Hasilnya juga akan disimpan dalam peta fitur yang berbeda. Namun pada akhirnya, semua peta fitur ini akan diambil dan digabungkan sebagai hasil akhir dari *convolution layer* [13].

Berikut adalah penjelasan beberapa unsur dalam *convolution layer* [16]

1. $[s]$ Stride merupakan pergeseran kernel pada tiap langkah. Stride berguna untuk mengurangi beban komputasi, semakin jauh pergeseran maka akan semakin banyak input yang dilewati
2. $[p]$ Padding merupakan layer tambahan untuk membuat ukuran input sama dengan output dan biasanya bernilai 0. Ukuran padding 1 artinya ada padding berukuran 1 pixel di setiap sisi input dimana jika f genap maka $p = \frac{f}{2}$ dan jika f ganjil maka $p = \frac{f-1}{2}$
3. $[f]$ Filter : number of channels output (Berupa ketebalan output)
4. $[n_c]$ Number of Channels : ketebalan image yaitu berupa jumlah warna pada image dengan rumus $n_c = f$
5. $[n]$ Kernel Size: ukuran image/layer. Jika kernel size sama dengan input size maka disebut *fully connected layer*.

Rumus menghitung ukuran output [17]:

$$n_{out} = \frac{n_{in} + 2p - f}{s} + 1$$

sehingga didapatkan,

$$\begin{array}{ccccc} \text{Input Image} & * & \text{Kernel} & = & \text{Output} \\ [n, n, n_c] & * & [f, f, n_c] & = & [n_{out}, n_{out}, f] \end{array}$$

Rumus menghitung entry:

- 1) *Overlay filter* ke *input*, lakukan *element-wise multiplication*, dan tambahkan hasilnya
- 2) Geser *overlay* sesuai dengan langkah stride, kemudian lakukan kalkulasi yang sama dengan langkah satu, dst

2.3.2.2 Max Pooling

Tujuan utama dari *Pooling Layer* adalah untuk mengurangi jumlah parameter dari tensor *input* sehingga dapat membantu mengurangi *overfitting*, ekstrak fitur representatif dari tensor *input*, mengurangi perhitungan dan dengan demikian membantu efisiensi [13].

Jenis operasi *pooling* yang paling sering digunakan pada CNN adalah *Max Pooling* yaitu mengambil nilai maksimum di setiap window/blok untuk mengurangi ukuran peta fitur yang hanya akan menyimpan informasi penting dan signifikan dalam proses klasifikasi nantinya. Dengan kata lain, Max Pooling akan meringkas feature map dengan mencari nilai atau value terbesar berdasarkan pergeseran window *pooling* [13]. Rumus Max Pooling output shape [18]:

$$\left\lfloor \frac{I_x - P}{s} \right\rfloor + 1$$

Keterangan:

I_x = ukuran x atau y

P = ukuran jendela Pooling
 s = stride

2.3.3 Komponen Klasifikasi

2.3.3.1 *Fully Connected*

Fully Connected layer adalah layer terakhir yang berada pada arsitektur CNN dimana lapisan ini menghubungkan semua neuron yang terhubung di seluruh lapisan sebelumnya dengan neuron di lapisan selanjutnya. Layer ini bertujuan untuk implementasi transformasi dimensi input data berbentuk *feature map* dengan tujuan agar dapat dilakukan klasifikasi secara linear secara relevan terhadap kelas-kelas yang ada [19]. Lapisan akan memiliki jumlah neuron output yang sama dengan jumlah kelas [20]. Output dari proses klasifikasi pada tahap ini dinyatakan pada persamaan sebagai berikut:

$$z_j = \sum_{i=1}^c w_{i,j}^T x_i + b_j$$

Keterangan

z_j = output

x_i = input dari jaringan

$w_{i,j}$ = matriks berisi bobot konektivitas antar unit dan jaringan (ukuran $i \times j$)

i = jumlah fitur dari input

j = jumlah dari kelas

b_j = bias vektor

2.3.3.2 *Softmax*

Softmax merupakan fungsi aktivasi pada layer output yang menjadi pembeda dengan *fully connected* yang menggunakan fungsi aktivasi ReLU. Fungsi aktivasi ini akan mentransformasi data input menjadi sebuah nilai probabilitas klasifikasi suatu kelas.

Peran *softmax* sangat penting karena *softmax* dapat mengubah skor menjadi distribusi probabilitas yang dinormalisasi yang mampu ditampilkan ke pengguna atau digunakan sebagai input ke sistem lain. Untuk alasan ini, biasanya ditambahkan fungsi *softmax* sebagai lapisan terakhir dari jaringan saraf [21,22]. Berikut merupakan persamaan dari fungsi aktivasi *softmax*:

$$p_i = \frac{e^{x_i + \log(C)}}{\sum_{k=1}^N e^{x_k + \log(C)}}$$

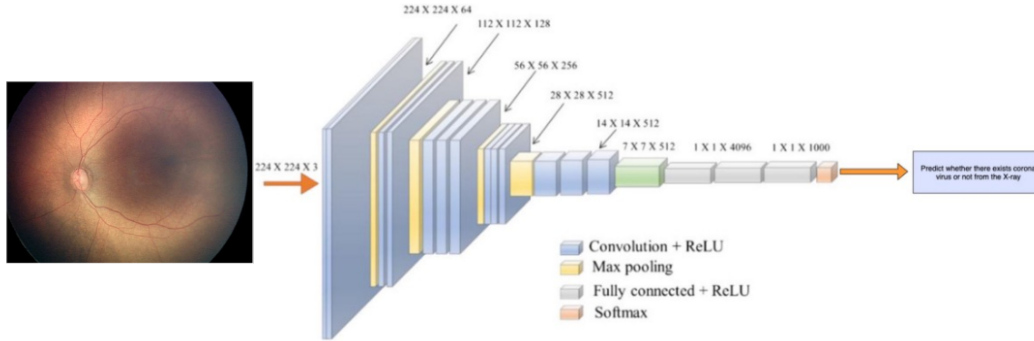
Keterangan

x_i = nilai strength neuron

$\log(C)$ = nilai konstan yang dapat ditentukan, secara umum = $\max(x)$

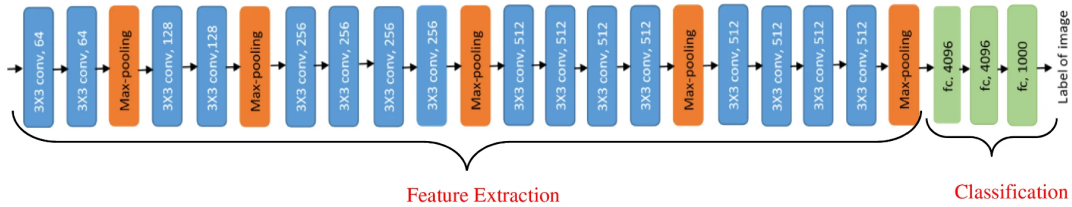
2.4 VGG-19

VGG-19 adalah arsitektur jaringan pada Convolution Neural Network dengan operasi multilayer. Model ini memiliki total 19 lapisan berbobot yang terdiri dari 16 *convolution layer* dan 3 *fully connected layer*. VGG-19 juga dilengkapi dengan 5 *max pooling layers* dan 1 *softmax layers*.



Gambar x. VGG-19 [23]

Digunakan konvolusi dengan filter (*convolutional layer*) berukuran 3x3 yang terletak pada lapisan atas untuk meningkatkan kedalaman level pada jaringan. Untuk mengurangi ukuran volume, digunakan *max pooling layers* dengan pool size sebesar 2x2. Digunakan pula 3 *fully connected layer* dengan nilai 4096, 4096, 1000. Serta, *size image* untuk input memiliki ukuran 224 x 224 piksel [24]. Detail struktur model VGG-19 dapat dilihat sebagai berikut.



Gambar x. Struktur Model VGG-19 [25]

2.5 Parameter

2.5.1 Optimizer

Optimizer adalah algoritma atau metode yang digunakan untuk mengubah atribut pada *neural network* seperti *weight* dan *learning rate* untuk mengurangi nilai *loss*. [26]

2.5.1.1 Adam Optimizer

Optimizer ADAM adalah algoritma optimalisasi berbasis gradien orde pertama untuk fungsi stokastik dengan tujuan memperbarui bobot secara iteratif yang didasarkan pada data training [27]. Berdasarkan Kingma et al., 2014, optimisasi ADAM efisien secara komputasi, memiliki sedikit kebutuhan memori, invarian terhadap penskalaan gradien diagonal, dan cocok untuk permasalahan yang besar dalam hal data/parameter. Rumus perhitungan optimasi ADAM ditunjukkan pada persamaan berikut;

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \cdot \hat{m}_t$$

θ_{t+1} : parameter hasil pembaruan
 θ_t : parameter hasil pembaruan sebelumnya
 η : learning rate
 \hat{m}_t : gradien kuadrat momen orde pertama
 \hat{v}_t : gradien kuadrat momen orde kedua
 ε : skalar kecil untuk mencegah pembagian dengan nol

2.5.2 Learning Rate

Learning rate adalah hyperparameter yang mengontrol seberapa banyak perubahan model dalam menanggapi kesalahan/error yang diperkirakan setiap kali bobot model diperbarui [28]. Dengan kata lain, *learning rate* mengontrol seberapa cepat model disesuaikan dengan masalah. Secara khusus, *learning rate* adalah hyperparameter yang dapat dikonfigurasi dimana digunakan dalam pelatihan *neural networks* yang memiliki nilai positif kecil dan nilainya sering berada dalam kisaran antara 0,0 dan 1,0.

2.5.3 Loss Function

Loss function adalah suatu fungsi yang digunakan untuk mengevaluasi algoritma dengan menghitung perbedaan antara output yang dikeluarkan oleh algoritma saat ini dengan output yang diharapkan. Jika prediksi model mengandung banyak kesalahan, *loss function* akan menghasilkan angka yang lebih tinggi [29].

2.5.3.1 Sparse Categorical Cross-Entropy

Akan digunakan *loss function sparse categorical cross-entropy* yang dipilih berdasarkan data yang kami olah. *Cross entropy* adalah *loss function* yang banyak digunakan pada tugas klasifikasi multi-class dan biner. *Cross entropy* mengukur perbedaan antara dua distribusi probabilitas untuk variabel acak tertentu atau serangkaian peristiwa. Tujuannya adalah untuk meminimalkan *loss*, yaitu semakin kecil *loss* semakin baik modelnya. Model yang sempurna memiliki *cross-entropy loss* 0. Lebih spesifik, *sparse categorical cross-entropy* digunakan sebagai *loss function* bagi klasifikasi *multi-class* dimana label dari tiap outputnya merupakan sebuah integer seperti (0,1,2,3,...) [30].

Cross-entropy dapat dihitung menggunakan probabilitas kejadian dari P dan Q, sebagai berikut:

$$H(P, Q) = - \sum_x^X P(x) \log(Q(x))$$

$H(P, Q)$: fungsi *cross-entropy*
 P : distribusi target
 Q : perkiraan distribusi target
 X : banyaknya kelas

2.5.4 Epoch

Epoch merupakan hyperparameter yang menentukan berapa kali algoritma pembelajaran akan bekerja mengolah seluruh dataset training. Seiring bertambahnya jumlah epoch, semakin banyak pula *weight* (bobot) yang berubah dalam Neural Network [31].

2.5.5 Batch Size

Batch Size adalah jumlah total data train yang ada dalam satu *batch*. Kita tidak dapat meneruskan seluruh dataset ke dalam jaringan secara sekaligus, sehingga kita harus membagi dataset menjadi jumlah *batch* [31].

2.5.6 Dropout Rate

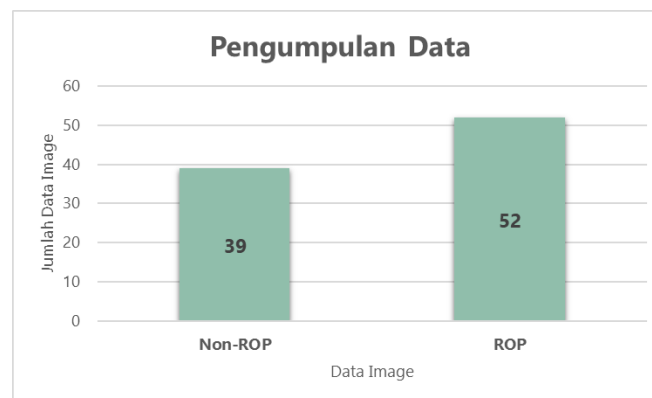
Dropout adalah penghapusan suatu node pada jaringan neural network yang mengurangi overfitting. Dropout rate adalah probabilitas pelatihan node tertentu pada suatu layer dimana jika dropout rate bernilai 1.0 artinya tidak ada dropout dan jika 0.0 artinya tidak ada output dari layer. Nilai optimal untuk hidden layer berada diantara 0.5-0.8 sedangkan pada layer input adalah 0.8 [32].

3. Data dan Rancangan Penelitian

3.1 Pengambilan Data

Data yang didapatkan melalui laman kaggle yaitu <https://www.kaggle.com/code/solennollivier/rop-2classclassification/data> berupa 91 citra fundus mata dengan format jpg berukuran 1.47-1.55 mb. Data-data tersebut berisi citra fundus mata yang tidak ROP, penderita ROP Stage 1, penderita ROP Stage 2, dan penderita ROP Stage 3.

Berikut visualisasi dari pengumpulan data yang dilakukan:

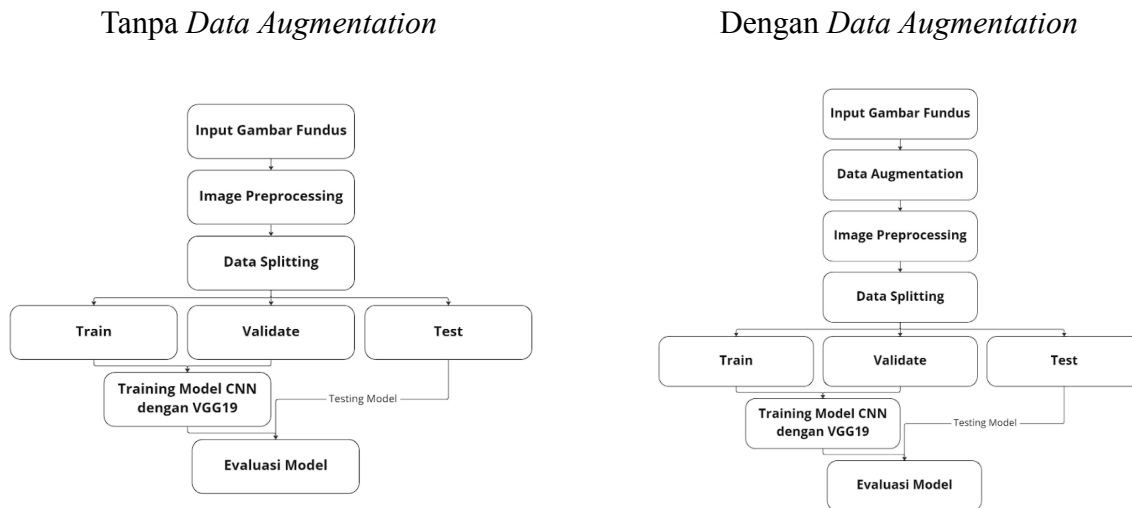


Dari grafik diatas, dapat terlihat bahwa pendistribusian jumlah data dari mata normal dan mata ROP tidak seimbang sehingga akan dilakukan proses augmentasi terlebih dahulu pada dataset sebelum data akan digunakan pada tahap selanjutnya.

Augmentasi data adalah suatu proses dalam pengolahan data gambar dengan mengubah atau memodifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun manusia masih dapat mengetahui bahwa gambar yang diubah tersebut adalah gambar yang sama [33]. Dari adanya keragaman data maka augmentasi dapat meningkatkan akurasi dari model CNN yang dilatih karena dengan augmentasi model mendapatkan data-data tambahan yang dapat berguna untuk membuat model yang dapat melakukan generalisasi dengan lebih baik [10].

3.2 Tahapan Penelitian

Langkah atau tahapan yang dilakukan pada penelitian ini adalah sebagai berikut.



4. Simulasi dan Analisis Hasil

4.1 Augmentasi Data

Melakukan *training* pada model dengan jumlah data yang kecil dapat menyebabkan *overfitting* saat proses *training*. Untuk menghindari masalah ini, maka kita akan membuat citra fundus mata yang baru dari dataset yang ada. Acuan nilai untuk augmentasi diperoleh berdasarkan penelitian terdahulu. Kemudian, dilakukan *trial and error* beberapa kali dengan menggunakan angka tertentu (yang tentunya masih dalam rentang acuan) untuk augmentasi data. Diperoleh angka parameter augmentasi yang menghasilkan model klasifikasi terbaik adalah *rotation_range* 2, *width_shift_range* 0.05, *height_shift_range* 0.05, *zoom_range* [0.85,1.15], dan *horizontal_flip*. Dari proses augmentasi data yang dilakukan, dihasilkan total 2124 data citra fundus mata yang akan digunakan ke proses selanjutnya.

4.2 Image Pre-processing

Untuk memudahkan dalam pengolahan gambar, langkah yang dapat dilakukan adalah mengubah ukuran citra (*image resizing*) dan mengubah kualitas citra (*image rescaling*). Pada *image resizing*, ukuran citra akan diubah menjadi 224 piksel x 224 piksel x 3 piksel agar sesuai dengan ukuran input arsitektur VGG-19. Setelah diubah ukurannya, akan dilakukan *rescaling* citra dengan membagi nilai setiap piksel dengan 255 sehingga dapat meringankan beban komputasi.

4.3 Data Splitting

Data splitting dilakukan untuk mencegah *overfitting*. Data splitting dilakukan dengan perbandingan 80% untuk Data Training, 10% untuk Data Validation, dan 10% untuk Data Testing [11]. Oleh karena itu, dari 91 citra fundus akan dilakukan data splitting menjadi 73 buah Data Training, 9 buah Data Validation, dan 9 buah Data Testing.

4.4 Evaluasi Kinerja Model

Dalam menilai kinerja suatu model, akan dilakukan perhitungan Accuracy, Precision, Recall, dan F1-Score [12]. Sebelum itu, kita harus mengetahui komponen yang ada di Confusion Matrix yang sudah diaplikasikan ke kasus ini.

		Actual Values	
		ROP	Non-ROP
Predicted Values	ROP	TP	FP
	Non-ROP	FN	TN

Terdapat 4 komponen pada Confusion Matrix, TP (True Positive) yaitu pasien menderita ROP dan dari model yang dibuat memprediksi pasien tersebut menderita ROP, TN (True Negative) yaitu pasien tidak menderita ROP dan dari model yang dibuat memprediksi pasien tersebut tidak menderita ROP, FP (False Positive) adalah pasien tidak menderita ROP tetapi dari model yang dibuat memprediksi pasien tersebut menderita ROP, dan FN (False Negative) adalah pasien menderita ROP tetapi dari model yang dibuat memprediksi pasien tersebut tidak menderita ROP.

Accuracy adalah rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Precision adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Recall adalah rasio benar positif dibandingkan dengan keseluruhan data benar positif dan F1-Score adalah harmonic mean dari precision dan recall.

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

$$Precision = \frac{TP}{TP+FP}.$$

$$Recall = \frac{TP}{TP+FN}.$$

$$F1 - Score = \frac{1}{F1} = \frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right)$$

4.3 Skenario Pengujian

Terdapat 2 model CNN (*Convolutional Neural Network*) yang dibangun dalam penelitian ini:

1. Model CNN (*Convolutional Neural Network*) menggunakan VGG-19 tanpa augmentasi data;
2. Model CNN (*Convolutional Neural Network*) menggunakan VGG-19 dengan augmentasi data.

Dari kedua model tersebut akan dilakukan pengujian terhadap model untuk mendapatkan nilai akurasi setiap model.

4.4 Hyperparameter

Setiap model yang digunakan diatur untuk memiliki *hyperparameter* berikut dengan nilai/tipe yang sama:

<i>Hyperparameters</i>	Nilai/Tipe untuk Model tanpa Augmentasi Data	Nilai/Tipe untuk Model dengan Augmentasi Data	Referensi
<i>Optimizer</i>	Adam	Adam	[20]
<i>Learning rate</i>	2×10^{-5}	2×10^{-5}	

<i>Hyperparameters</i>	Nilai/Tipe untuk Model tanpa Augmentasi Data	Nilai/Tipe untuk Model dengan Augmentasi Data	Referensi
Fungsi aktivasi	<i>Rectified Linear Unit</i> (ReLU)	<i>Rectified Linear Unit</i> (ReLU)	
Loss Function	Sparse Categorical Cross-Entropy	Sparse Categorical Cross-Entropy	
<i>Batch size</i>	10	32	Disesuaikan dengan kapasitas kemampuan komputasi.
<i>Epoch</i>	20	20	

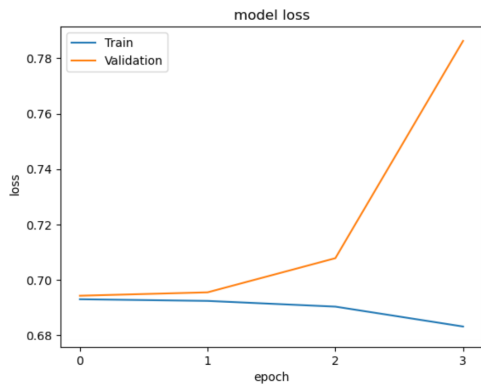
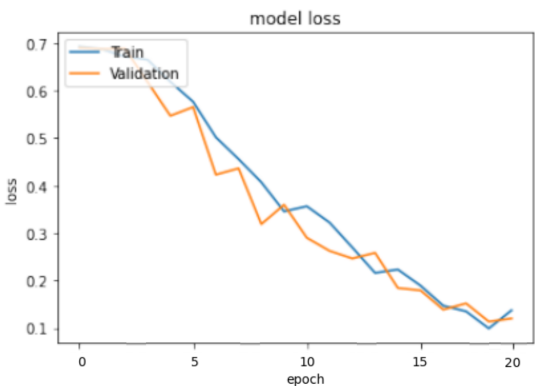
Tabel 1. *Hyperparameter*

4.5. Hasil Pengujian

Didapat hasil pengujian dari kedua model sebagai berikut:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
A1	70%	30%	50%	37%
A2	96%	96%	96%	96%

Tabel 2. Evaluasi Model

Model	A1	A2
Grafik Loss		

Tabel 3. Grafik Loss Function dari Model

Keterangan:

1. A1 : Model CNN menggunakan VGG-19 tanpa augmentasi data;
2. A2 : Model CNN menggunakan VGG-19 dengan augmentasi data.

Dari hasil pengujian diatas, maka didapatkan model terbaik untuk melakukan pendeteksian ROP (*Retinopathy of Prematurity*) pada bayi prematur adalah model CNN menggunakan VGG-19 dengan augmentasi data, dengan 96% accuracy, 96% precision, 96% recall, 96% F1-Score, dan 0.14 Loss.

5. Penutup

5.1. Kesimpulan

Dari hasil pengujian bab sebelumnya, tim peneliti menyimpulkan:

1. Cara kerja metode CNN dalam mendeteksi *Retinopathy of prematurity* (ROP) pada bayi prematur adalah dari dataset yang ada dilakukan *labeling* data untuk mengetahui data mana yang ROP dan mana yang non-ROP. Kemudian, dilakukan *training* data dengan menggunakan CNN. Dari *labeling* data tersebut terbentuk suatu model yang kemudian dapat digunakan untuk mengklasifikasikan data mata baru ke dalam kelas yang sudah ditentukan yaitu ROP atau non-ROP.
2. Didapatkan analisis kinerja model dari metode *Convolutional Neural Network* dalam mendeteksi *Retinopathy of prematurity* (ROP) pada bayi prematur untuk model CNN menggunakan VGG-19 dengan augmentasi data sebesar 96% accuracy, 96% precision, 96% recall, 96% F1-Score, dan 0.14 Loss. Sedangkan untuk model CNN menggunakan VGG-19 tanpa augmentasi data sebesar 60% accuracy, 30% precision, 50% recall, 37% F1-Score, dan 0.67 Loss.
3. Pengaruh penggunaan augmentasi data dalam mendeteksi *Retinopathy of prematurity* (ROP) pada bayi prematur dapat dilihat pada perbandingan evaluasi model tanpa augmentasi dan model dengan augmentasi. Didapatkan akurasi untuk model tanpa augmentasi data sebesar 60% sedangkan untuk model dengan augmentasi data diperoleh sebesar 96%. Artinya, model dengan data augmentasi sangat berpengaruh dalam memberikan evaluasi model yang lebih baik yang menyebabkan pengaruh pada pendeteksian *Retinopathy of prematurity* (ROP) pada bayi prematur.

5.2. Saran

Untuk penelitian di kemudian hari, peneliti menyarankan untuk menggunakan *multi-class classification* dalam melakukan klasifikasi *Retinopathy of prematurity* (ROP) pada mata bayi prematur agar dapat mengklasifikasikan ROP berdasarkan *stage* yang ada yaitu apakah mata non-ROP, mengidap ROP *stage* 1, ROP *stage* 2, ROP *stage* 3, ROP *stage* 4, atau ROP *stage* 5. Untuk itu, diperlukan juga data yang lebih lengkap dan beragam, yang memuat mata ROP pada setiap *stage*.

DAFTAR PUSTAKA

- [1] L. K. Anna, "Indonesia urutan Kelima Jumlah kelahiran prematur," KOMPAS.com, 28-Apr-2015. [Online]. Available: <https://health.kompas.com/read/2015/04/28/151500923/Indonesia.Urutan.Kelima.Jumlah.Kelahiran.Prematur>. [Accessed: 22-Sep-2022]
- [2] G. Solahuddin, "Indonesia urutan 5 Jumlah Bayi prematur terbanyak di Dunia, Risiko ROP," Grid Health, 09-Jan-2019. [Online]. Available: <https://health.grid.id/read/351597676/indonesia-urutan-5-jumlah-bayi-prematur-terbanyak-di-dunia-risiko-rop?page=all>. [Accessed: 22-Sep-2022]
- [3] U. Irianto and D. Y. Boyoh, "GAMBARAN USIA KELAHIRAN KURANG DARI 35 MINGGU TERHADAP ANGKA KEJADIAN RETINOPATHY OF PREMATURITY DI RS XX JAKARTA SELATAN," Jurnal Kesehatan Surya Nusantara, vol. 10, no. 1, 2022.
- [4] W. Yahya, "Retinopati Prematuritas : Diagnosis dan Tatalaksana," Cermin Dunia Kedokteran, vol. 47, no. 10, p. 676, 2020.
- [5] I. Kurniawan, "Artificial Intelligence (AI) vs. Machine Learning vs. Deep Learning," School of Information Systems, 04-Mar-2020. [Online]. Available: <https://sis.binus.ac.id/2020/03/04/artificial-intelligence-ai-vs-machine-learning-vs-deep-learning/>. [Accessed: 06-Oct-2022].
- [6] Müller Andreas C. and S. Guido, Introduction to machine learning with python: A guide for data scientists. Sebastopol, CA: O'Reilly Media, Inc., 2018.
- [7] I. B. M. By: IBM Cloud Education, "What is Deep Learning?," IBM, 01-May-2020. [Online]. Available: <https://www.ibm.com/cloud/learn/deep-learning>. [Accessed: 06-Oct-2022].
- [8] Cholissodin, Imam & Soebroto, Arief. (2019). Buku Ajar AI, Machine Learning & Deep Learning. from 348003841_Buku_Ajar_AI_Machine_Learning_Deep_Learning
- [9] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, vol. 521, no. 7553, pp. 436-444, 2015. Available: 10.1038/nature14539.
- [10] E. Satria, "PENERAPAN DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR RESNET UNTUK KLASIFIKASI BATIK," Universitas Islam Negeri Sultan Syarif Kasim Riau Repository, 22-Jan-2021. [Online]. Available: <http://repository.uin-suska.ac.id/42565/>. [Accessed: 04-Nov-2022].
- [11] P. Baheti, "Train test validation split: How to & best practices [2022]," V7, 21-Oct-2022. [Online]. Available: <https://www.v7labs.com/blog/train-validation-test-set>. [Accessed: 13-Dec-2022].
- [12] H. N. B, "Confusion matrix, accuracy, precision, recall, F1 score," Medium, 01-Jun-2020. [Online]. Available: <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>. [Accessed: 13-Dec-2022].
- [13] "Pengertian dan Cara Kerja algoritma Convolutional Neural Network (CNN)," Trivusi. [Online]. Available: <https://www.trivusi.web.id/2022/04/algoritma-cnn.html>. [Accessed: 29-Oct-2022].
- [14] J. Brownlee, "A gentle introduction to the rectified linear unit (ReLU)," Machine Learning Mastery, 20-Aug-2020. [Online]. Available: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>. [Accessed: 14-Oct-2022].
- [15] A. Sheth, A. Shrivastava, and A. Sinhal, Intelligent Systems: Proceedings of SCIS 2021. SPRINGER, 2021.
- [16] Anak AI. "Mengenal Convolutional Neural Network (CNN)". 09-08-2020. [Youtube]. Available: <https://youtu.be/3NwE3Eu8g7c>. [Accessed: 29-Okt-2022].
- [17] P. Skalski, "Gentle Dive Into Math Behind Convolutional Neural Networks," Towards Data Science, 12-Apr-2019. [Online]. Available: <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9>. [Accessed: 29-Oct-2022].

- [18] O. W. Layton, "CS343: Neural networks - colby college." [Online]. Available: <https://cs.colby.edu/courses/F19/cs343/lectures/lecture11/Lecture11Slides.pdf>. [Accessed: 29-Oct-2022].
- [19] A. Kholik, "KLASIFIKASI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA TANGKAPAN LAYAR HALAMAN INSTAGRAM", vol. 2, no. 2, pp.10-20, 2021.
- [20] Y.-P. Huang, S. Vadloori, H.-C. Chu, E. Y.-C. Kang, W.-C. Wu, S. Kusaka, and Y. Fukushima, "Deep Learning Models For Automated Diagnosis of Retinopathy of Prematurity in preterm infants," Electronics, vol. 9, no. 9, p. 1444, 2020.
- [21] A. Yusuf, R. C. Wihandika, and C. Dewi, Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network, vol. 3, pp. 10595–10604, 2019.
- [22] T. Wood, "Softmax Function," DeepAI, 17-May-2019. [Online]. Available: <https://deeppai.org/machine-learning-glossary-and-terms/softmax-layer>. [Accessed: 29-Oct-2022].
- [23] S. Sharma, "How to train CNNs on ImageNet," Medium, 24-May-2020. [Online]. Available: <https://towardsdatascience.com/how-to-train-cnns-on-imagenet-ab8dd48202a9>. [Accessed: 29-Oct-2022].
- [24] M. Mateen, J. Wen, Nasrullah, S. Song, and Z. Huang, "Fundus image classification using VGG-19 architecture with PCA and SVD," Symmetry, vol. 11, no. 1, pp. 1–12, Dec. 2018.
- [25] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using VGG19: Caltech-101 Image Data Set," Journal of Ambient Intelligence and Humanized Computing, 2021.
- [26] S. Doshi, "Various Optimization Algorithms For Training Neural Network," Medium, 03-Aug-2020. [Online]. Available: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>. [Accessed: 14-Oct-2022].
- [27] S. Tilawah, "Adam Optimizer," Medium, 31-May-2020. [Online]. Available: <https://medium.com/@saritilawah9/adam-optimizer-80cc267522af>. [Accessed: 14-Oct-2022].
- [28] Brownlee, J. "Understand the Impact of Learning Rate on Neural Network Performance", Machine Learning Mastery. 25-January-2019. [Online]. Available: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>
- [29] Trivusi. "Mengenal 6 Jenis Loss Function pada Machine Learning", Trivusi. 17-September-2022. [Online]. Available: <https://www.trivusi.web.id/2022/08/loss-function.html>
- [30] Keras - categorical cross entropy loss function. Data Analytics. Retrieved December 13, 2022, from <https://vitalflux.com/keras-categorical-cross-entropy-loss-function/>
- [31] S. Sharma, "Epoch vs Batch Size vs Iterations," Towards Data Science, 23-Sep-2017. [Online]. Available: <https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9>. [Accessed: 03-Nov-2022].
- [32] J. Brownlee, "A Gentle Introduction to Dropout for Regularizing Deep Neural Networks," Machine Learning Mastery, 06-Aug-2019. [Online]. Available: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>. [Accessed: 03-Nov-2022].
- [33] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning, 2017.