



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



RUMUSAN MASALAH

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?

TUJUAN

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 model dengan augmentasi data untuk memperoleh model terbaik dalam penerapan metode CNN terhadap kasus pendeteksian Retinopathy of Prematurity (ROP) pada bayi prematur.
3. Mengetahui pengaruh augmentasi data pada model dalam penerapan metode CNN terhadap kasus pendeteksian Retinopathy of Prematurity (ROP) pada bayi prematur.

BATASAN MASALAH

- Data yang dipakai adalah data citra fundus mata.
- Diberikan citra fundus mata yang mengalami ROP dan tidak mengalami ROP.
- Data image fundus mata yang mengalami ROP hanya dari stage 1 – 3.
- Arsitektur CNN yang digunakan adalah VGG-19.

PENELITIAN TERDAHULU

“Deep Learning Models for Automated Diagnosis of Retinopathy of Prematurity in Preterm Infants”



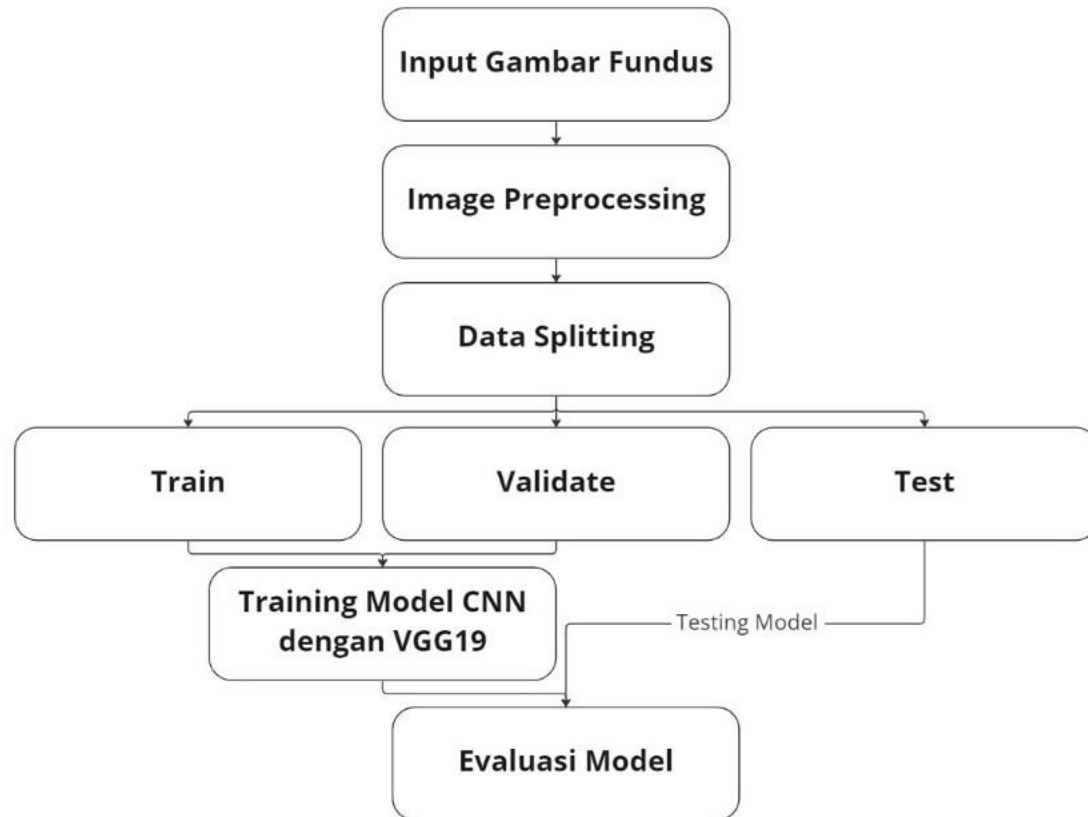
Perbandingan beberapa metode arsitektur CNN dalam mendiagnosa penyakit Retinopathy of Prematurity (ROP) pada bayi prematur dengan 109 citra fundus mata yang didapat dari neonatal intensive care units of Chang Gung Memorial Hospital, Linkou, Taiwan, dan Osaka Women's and Children's Hospital, Japan.

Classification Models	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)
VGG19	96.0	96.6	95.2	95.2
VGG16	88.1	96.6	76.2	94.1
InceptionV3	72.3	94.9	40.5	85.0
DenseNet	76.2	67.8	88.1	66.1
MobileNet	86.1	86.4	85.7	81.8

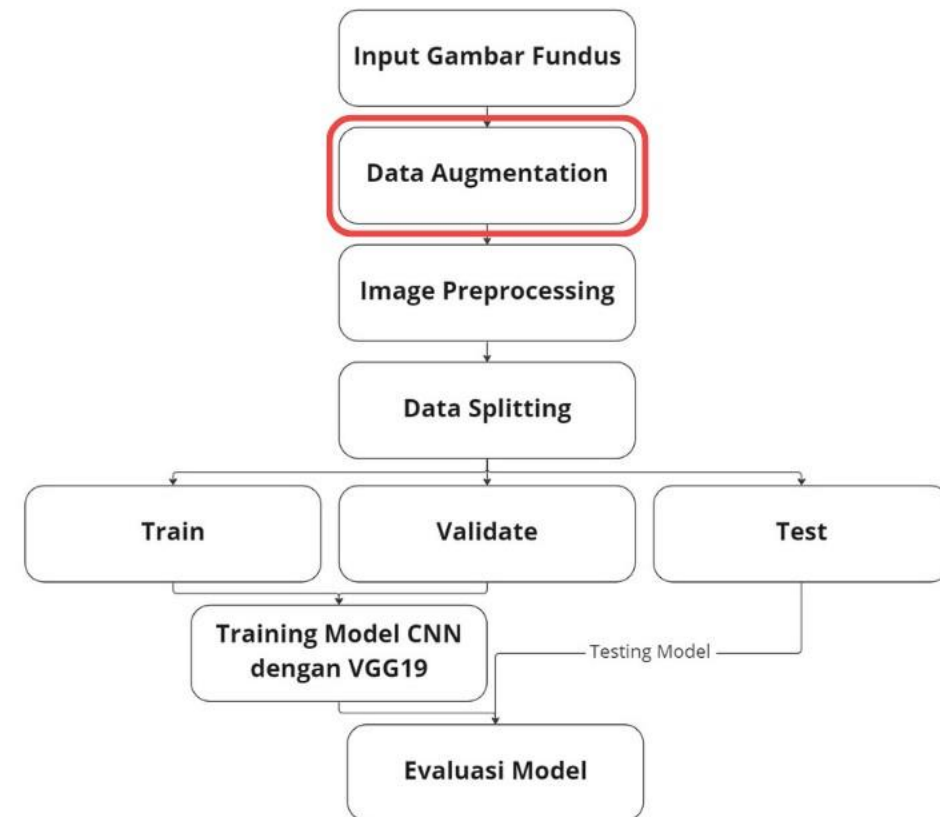
METODOLOGI PENELITIAN



Tanpa Data Augmentation



Dengan Data Augmentation



PENGUMPULAN DATA

- **Jenis Data**

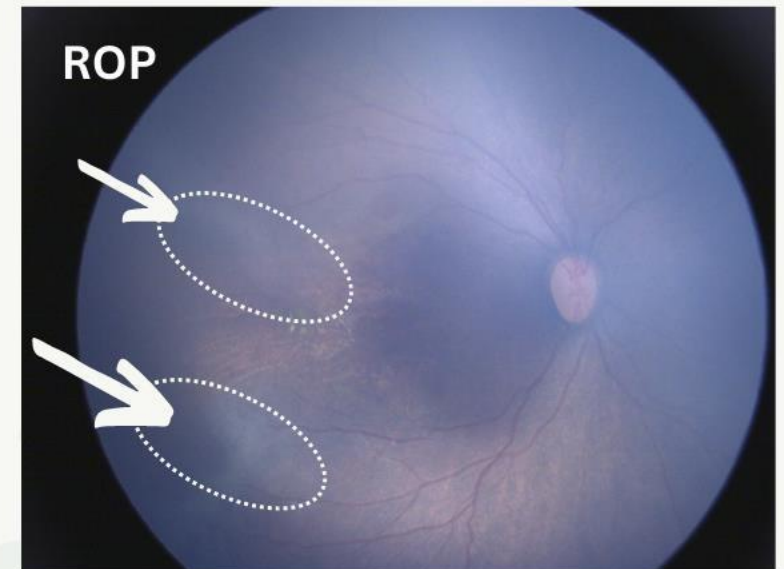
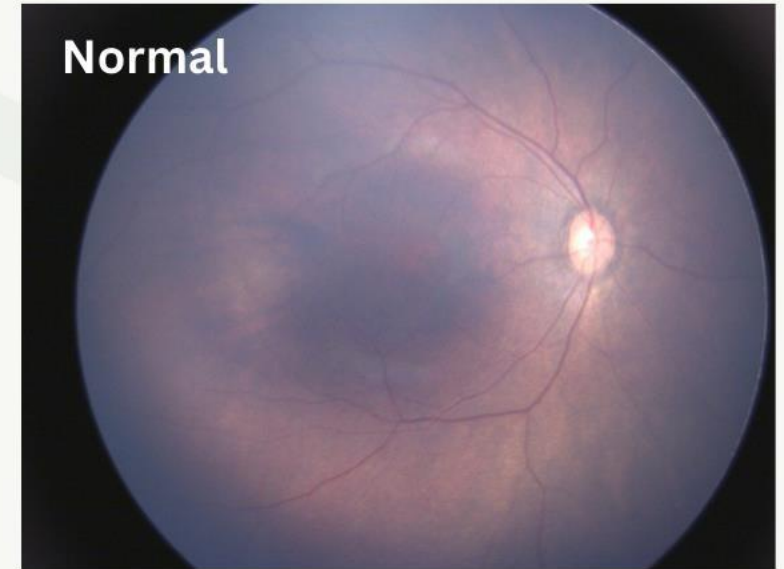
Citra fundus mata dengan format **.jpg**
berukuran **1.47-1.55 MB**.

- **Jumlah Sampel**

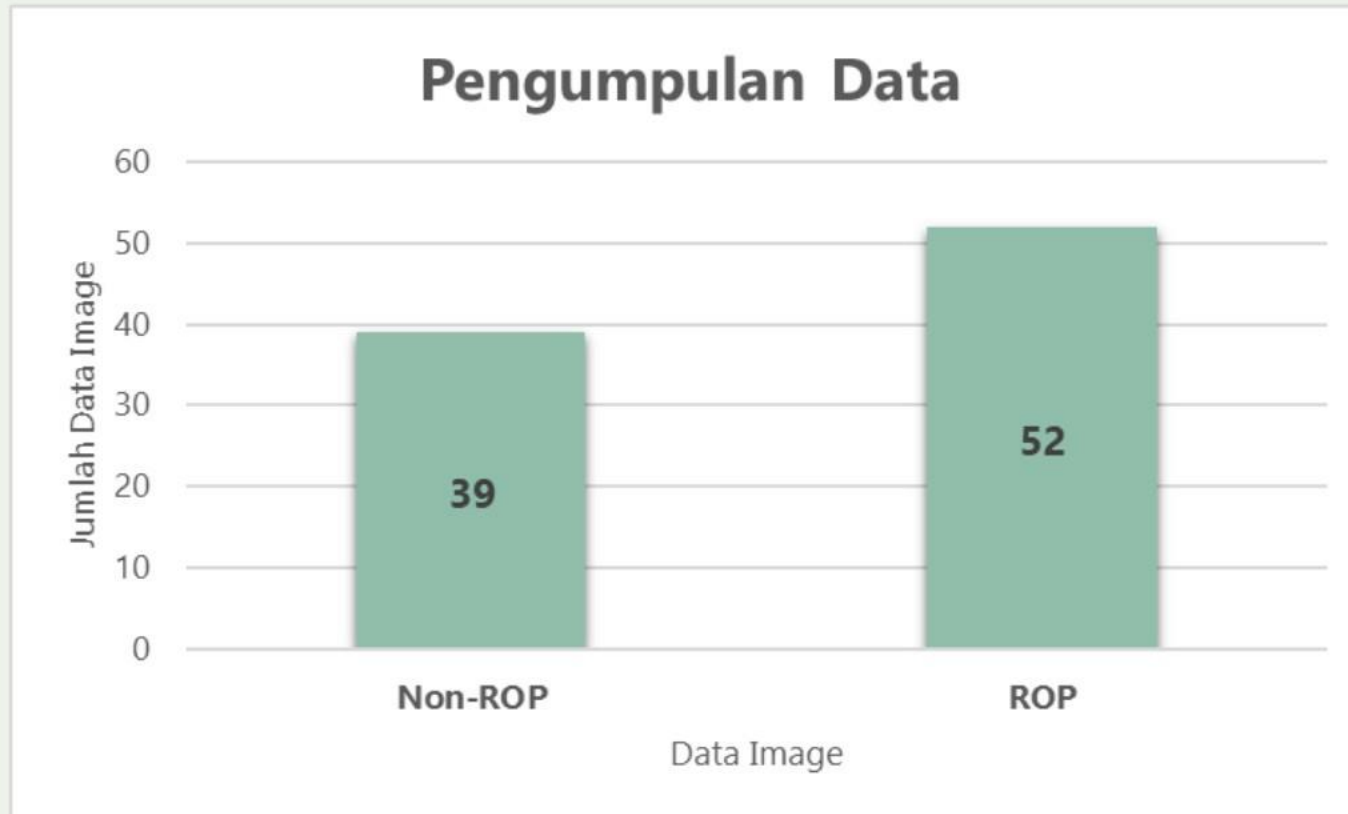
91 citra fundus mata yang terbagi menjadi
Non-ROP, Stage 1, Stage 2, dan Stage 3.

- **Sumber Data**

<https://www.kaggle.com/code/solennollivier/rop-2classclassification/data>.



DISTRIBUSI DATA



Distribusi jumlah data dari mata normal dan mata ROP

TIDAK SEIMBANG

Sehingga, akan dilakukan percobaan 2 model untuk model tanpa dan dengan

AUGMENTASI

terlebih dahulu pada dataset

PENELITIAN TERDAHULU



VGG-19

OPTIMIZER

ADAM

Adaptive Moment Estimation

LEARNING RATE

2×10^{-5}

LOSS FUNCTION

**Categorical
Cross-Entropy**

BATCH SIZE

10

PENELITIAN TERDAHULU



DATA AUGMENTATION

ROTATION RANGE

$[-3, 3]$

FLIP

HORIZONTAL

WIDTH SHIFT RANGE

$[-0.1, 0.1]$

HEIGHT SHIFT RANGE

$[-0.1, 0.1]$

ZOOM RANGE

$[0.85, 1.15]$

• Loading Data Citra pada Program (tanpa Augmentasi)



Import Modul & Package

Ambil lokasi directory data

Loading data

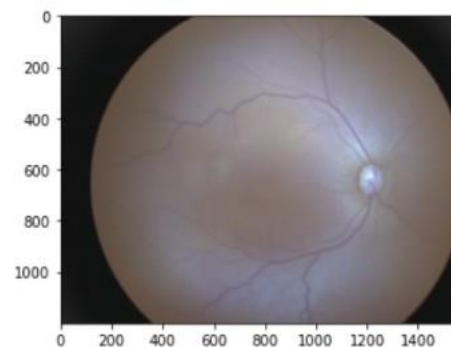
Cuplikan data

```
In [1]: # import modul yang dipakai.  
import tensorflow as tf  
import os  
import cv2  
import matplotlib.pyplot as plt  
import numpy as np
```

```
In [31]: # data directory  
data_dir = "C:/Users/User/Pemodelan (ROP)\Data"
```

```
In [32]: categories = ['ROP', 'NonROP']
```

```
In [33]: for category in categories:  
    path = os.path.join(data_dir, category)  
    for img in os.listdir(path):  
        img_array = cv2.imread(os.path.join(path, img))  
        plt.imshow(img_array)  
        plt.show()  
        break  
    break  
print('\n img size = ', img_array.shape)
```



img size = (1200, 1600, 3)

• Loading Data Citra pada Program (dengan Augmentasi)



Import Modul & Package

Ambil lokasi directory data

Kriteria Data Augmentation

Loading Data

input ke Directory Baru

```
1 # import modul yang dipakai.
2 import tensorflow as tf
3 import os
4 import cv2
5 import matplotlib.pyplot as plt
6 import numpy as np
7 from keras.preprocessing.image import ImageDataGenerator
8 from tensorflow.keras.utils import img_to_array
9 from tensorflow.keras.utils import array_to_img
10 from tensorflow.keras.utils import load_img
11 from skimage import io
```

```
1 data_dir = 'dataRETINA'
2 datagen = ImageDataGenerator(
3     rotation_range= 2,
4     width_shift_range= 0.05,
5     height_shift_range= 0.05,
6     zoom_range = [0.85, 1.15],
7     horizontal_flip = True,
8     fill_mode = 'nearest')
9
10 categories_non = ['NonROP']
11
12 for category in categories_non:
13     path = os.path.join(data_dir, category)
14     for img in os.listdir(path):
15         image = load_img('dataRETINA/NonROP/{0}'.format(img))
16         x = img_to_array(image)
17         x = x.reshape((1,) + x.shape)
18
19         i = 0
20         for batch in datagen.flow(x, batch_size=1,
21                                   save_to_dir='dataRETINAaugmen
22                                     i += 1
23                                     if i > 26:
24                                         break
```

Non-ROP
Data

```
1 data_dir = 'dataRETINA'
2 datagen = ImageDataGenerator(
3     rotation_range= 2,
4     width_shift_range= 0.05,
5     height_shift_range= 0.05,
6     zoom_range = [0.85, 1.15],
7     horizontal_flip = True,
8     fill_mode = 'nearest')
9
10 categories_sick = ['ROP']
11
12 for category in categories_sick:
13     path = os.path.join(data_dir, category)
14     for img in os.listdir(path):
15         image = load_img('dataRETINA/ROP/{0}'.format(img))
16         x = img_to_array(image)
17         x = x.reshape((1,) + x.shape)
18
19         i = 0
20         for batch in datagen.flow(x, batch_size=1,
21                                   save_to_dir='dataRETINAaugmen
22                                     i += 1
23                                     if i > 20:
24                                         break
```

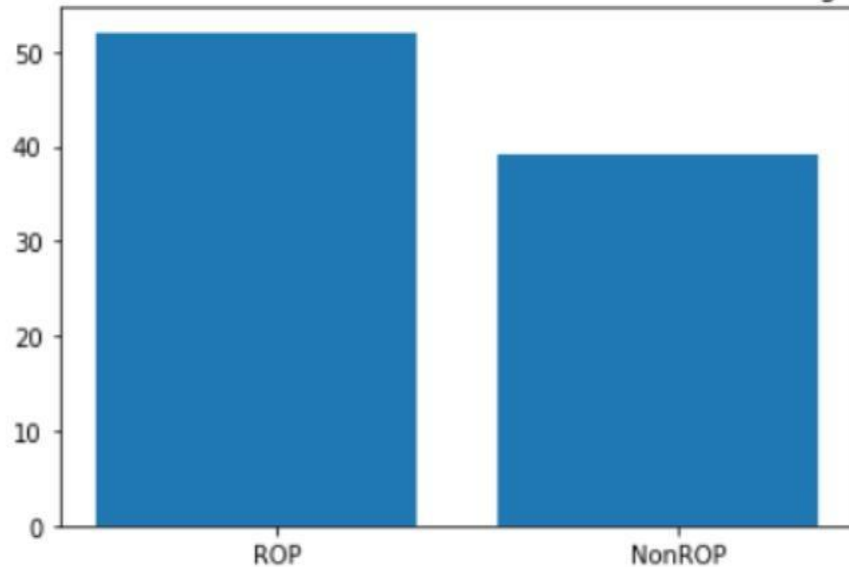
ROP
Data

• Perbandingan Jumlah Data



Model Tanpa Augmentasi

Distribusi Data ROP dan NonROP Pada Dataset SEBELUM Augmentasi

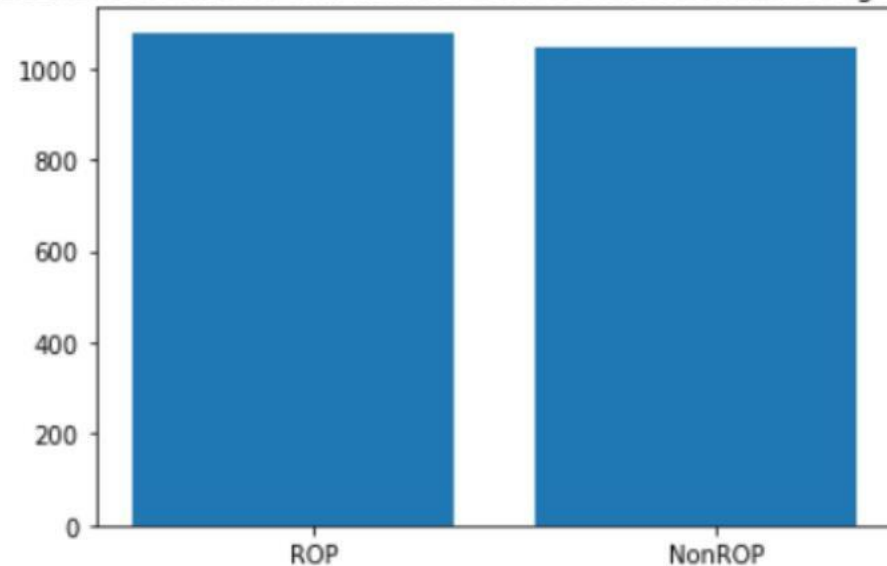


Jumlah data ROP pada dataset SEBELUM augmentasi: 52

Jumlah data NonROP pada dataset SEBELUM augmentasi: 39

Model Dengan Augmentasi

Distribusi Data ROP dan NonROP Pada Dataset SETELAH Augmentasi



Jumlah data ROP pada dataset SETELAH augmentasi: 1078

Jumlah data NonROP pada dataset SETELAH augmentasi: 1046

• Resize & Mengelompokkan Data



1

Resize citra fundus mata menjadi 224 x 224

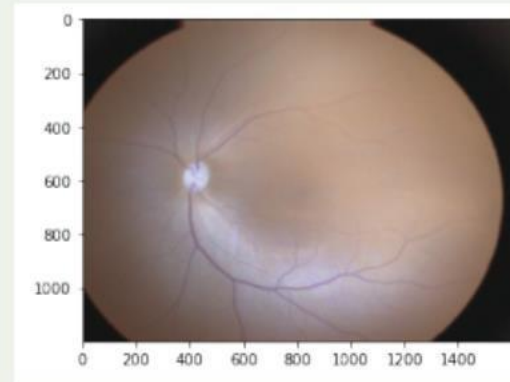


image size = (1200, 1600, 3)

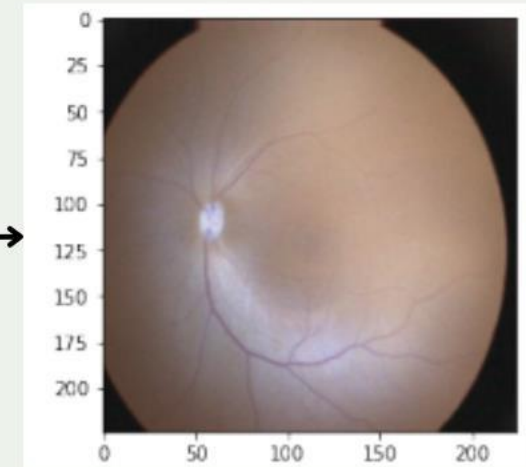


image size = (224, 224, 3)

2

Mengelompokkan data



Indeks 0 = ROP
Indeks 1 = Non-ROP

• Menampilkan Hasil Sample Data



1. Melakukan pengacakan data (shuffling data)

a. Model **Tanpa** Augmentasi

Data yang dipunya: [0, 0, 0, 0, 0, 0, 0, 0, 0]

Data yang dipunya setelah shuffle: [1, 1, 1, 0, 1, 1, 0, 1, 0]

b. Model **Dengan** Augmentasi

Data yang dipunya: [0, 0, 0, 0, 0, 0, 0, 0, 0]

Data yang dipunya setelah shuffle: [0, 1, 0, 0, 0, 0, 1, 0, 0]

2. Melakukan normalisasi dengan membagi tiap elemen dengan 255

3. Melakukan data splitting dengan rasio 8:1:1 [train:test:val]

Didapatkan jumlah data:

a. Model **Tanpa** Augmentasi

Data Train : 72

Data Test : 10

Data Validation : 9

b. Model **Dengan** Augmentasi

Data Train : 1699

Data Test : 213

Data Validation : 212

• Distribusi Hasil Splitting Data



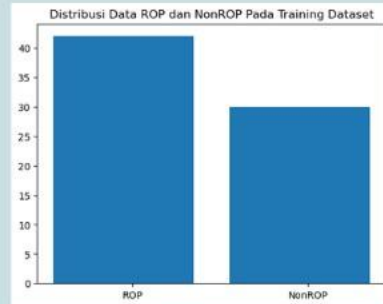
+
15

Training Dataset
[80%]

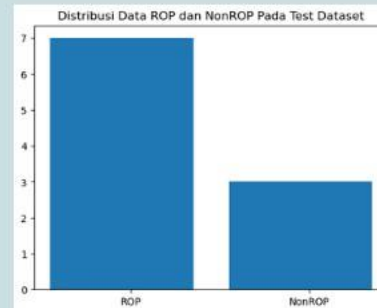
Test Dataset
[10%]

Validation Dataset
[10%]

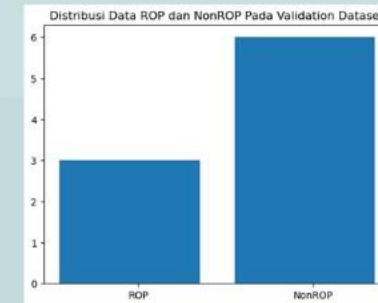
Model **Tanpa**
Augmentasi



ROP : 42
NonROP : 30



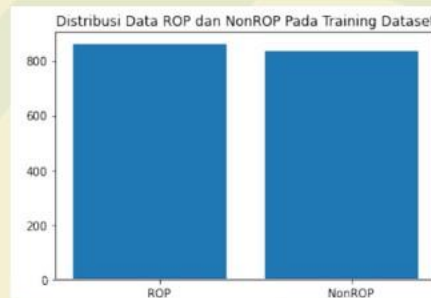
ROP : 7
NonROP : 3



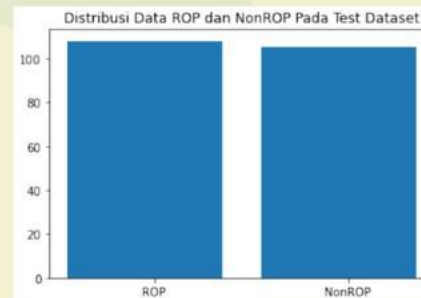
ROP : 3
NonROP : 6

Jumlah Data
Data Train : 72
Data Test : 10
Data Validation : 9

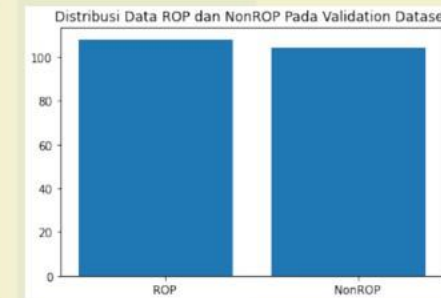
Model **Dengan**
Augmentasi



ROP : 849
NonROP : 850



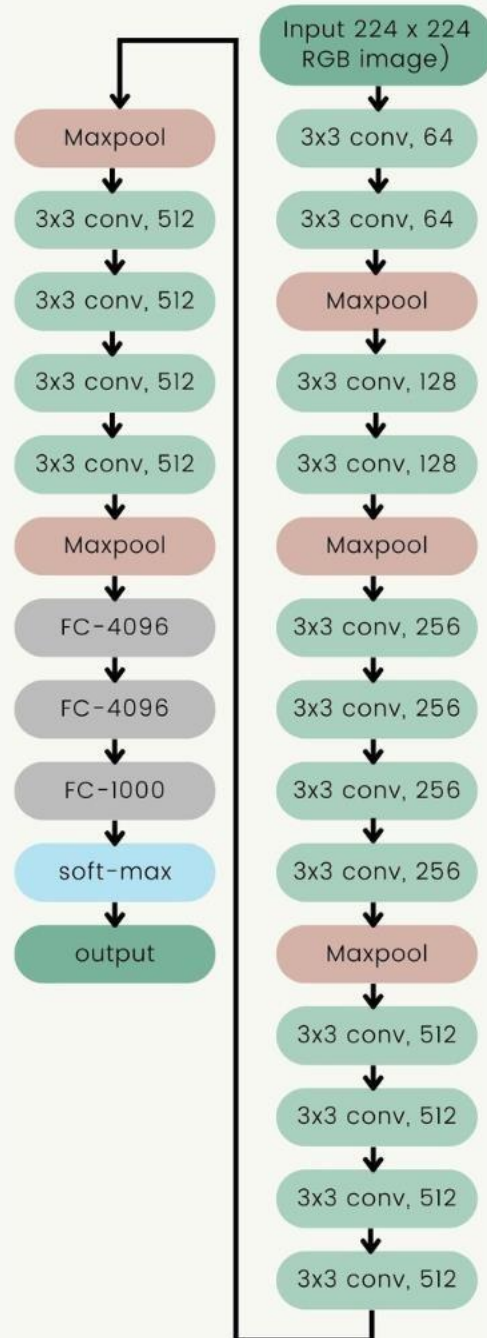
ROP : 113
NonROP : 99



ROP : 116
NonROP : 97

Jumlah Data
Data Train : 1699
Data Test : 213
Data Validation : 212

• Coding CNN: VGG-19



Dengan menerapkan parameter

Model **Tanpa** Augmentasi

- Optimizer : Adam
- Learning rate : 2×10^{-5}
- Loss function : Sparse Categorical Cross-Entropy
- Batch size : 10
- Epoch : 20

Model **Dengan** Augmentasi

- Optimizer : Adam
- Learning rate : 2×10^{-5}
- Loss function : Sparse Categorical Cross-Entropy
- Batch size : 32
- Epoch : 20

• Model Summary CNN: VGG-19



Model: "sequential_1"

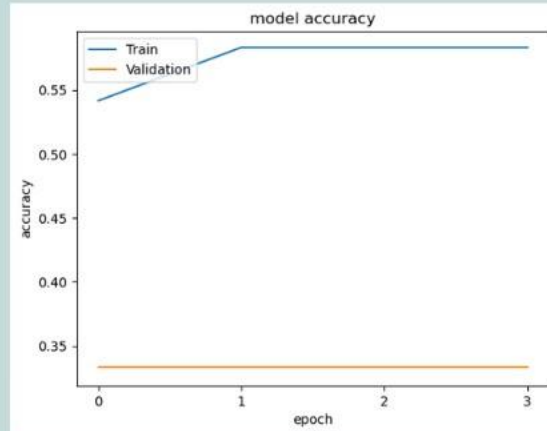
Layer (type)	Output Shape	Param #
conv2d_16 (Conv2D)	(None, 224, 224, 64)	1792
conv2d_17 (Conv2D)	(None, 224, 224, 64)	36928
max_pooling2d_5 (MaxPooling 2D)	(None, 112, 112, 64)	0
conv2d_18 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_19 (Conv2D)	(None, 112, 112, 128)	147584
max_pooling2d_6 (MaxPooling 2D)	(None, 56, 56, 128)	0
conv2d_20 (Conv2D)	(None, 56, 56, 256)	295168
conv2d_21 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_22 (Conv2D)	(None, 56, 56, 256)	590080
conv2d_23 (Conv2D)	(None, 56, 56, 256)	590080
max_pooling2d_7 (MaxPooling 2D)	(None, 28, 28, 256)	0
conv2d_24 (Conv2D)	(None, 28, 28, 512)	1180160
conv2d_25 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_26 (Conv2D)	(None, 28, 28, 512)	2359808
conv2d_27 (Conv2D)	(None, 28, 28, 512)	2359808

max_pooling2d_8 (MaxPooling 2D)	(None, 14, 14, 512)	0
conv2d_28 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_29 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_30 (Conv2D)	(None, 14, 14, 512)	2359808
conv2d_31 (Conv2D)	(None, 14, 14, 512)	2359808
max_pooling2d_9 (MaxPooling 2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_4 (Dense)	(None, 4096)	102764544
dropout_2 (Dropout)	(None, 4096)	0
dense_5 (Dense)	(None, 4096)	16781312
dropout_3 (Dropout)	(None, 4096)	0
dense_6 (Dense)	(None, 1000)	4097000
dense_7 (Dense)	(None, 2)	2002
=====		
Total params: 143,669,242		
Trainable params: 143,669,242		
Non-trainable params: 0		

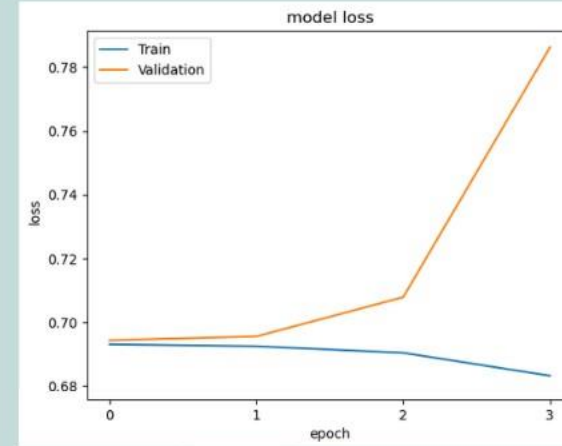
• Evaluasi Model: Model Accuracy & Loss



Model **Tanpa**
Augmentasi

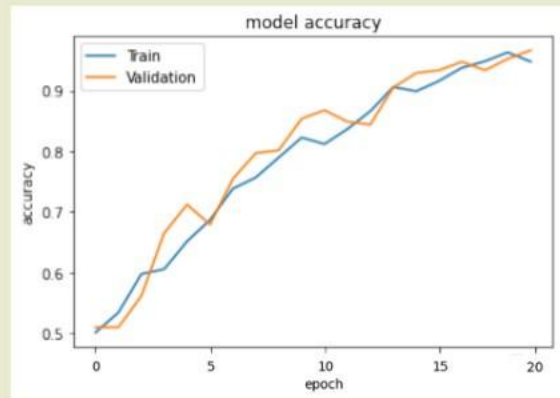


Diperoleh akurasi sebesar **70%**

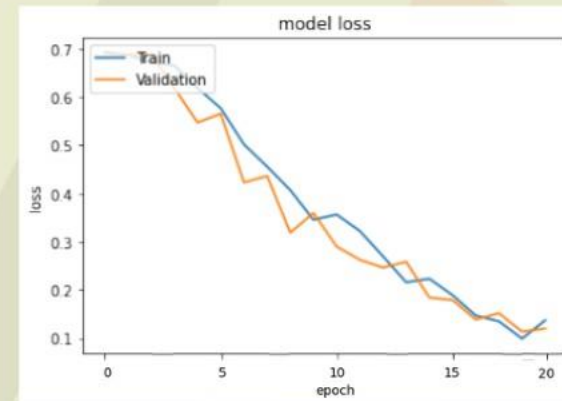


Diperoleh loss sebesar **0.62**

Model **Dengan**
Augmentasi



Didapatkan akurasi sebesar **96%**



Didapatkan loss sebesar **0.14**

• Evaluasi Model: Classification Report



Accuracy : rasio prediksi benar (positif dan negatif) dengan keseluruhan data

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

Precision : Perbandingan antara True Positive dengan banyaknya data yang diprediksi positif

$$precision = \frac{TP}{TP + FP}$$

Recall : Perbandingan antara True Positive dengan banyaknya data yang sebenarnya positif

$$recall = \frac{TP}{TP + FN}$$

F1-score : Harmonic mean dari precision dan recall

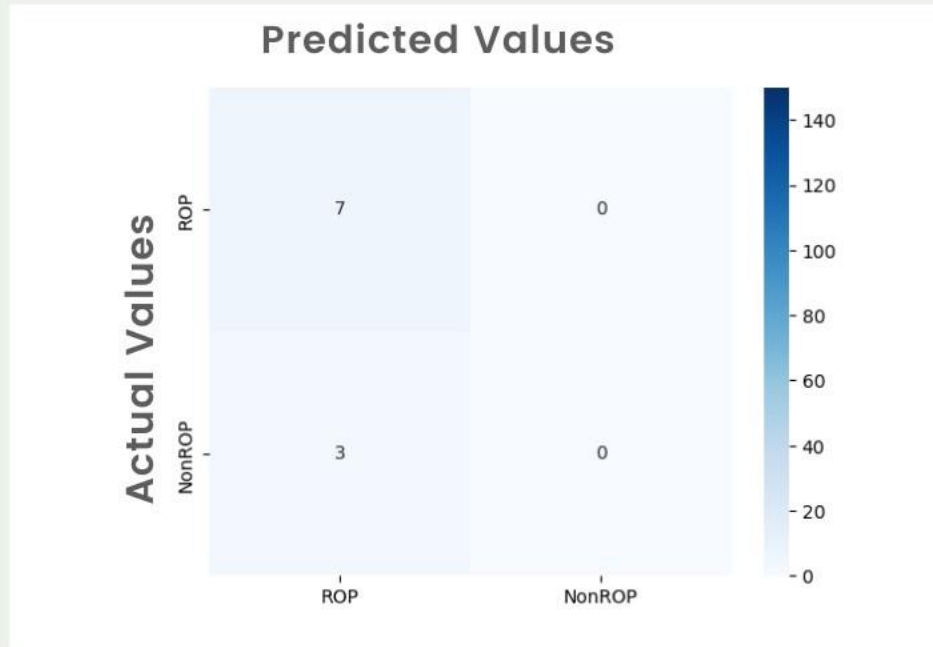
$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right)$$

Support : Jumlah data test yang digunakan

• Evaluasi Model: Confusion Matrix



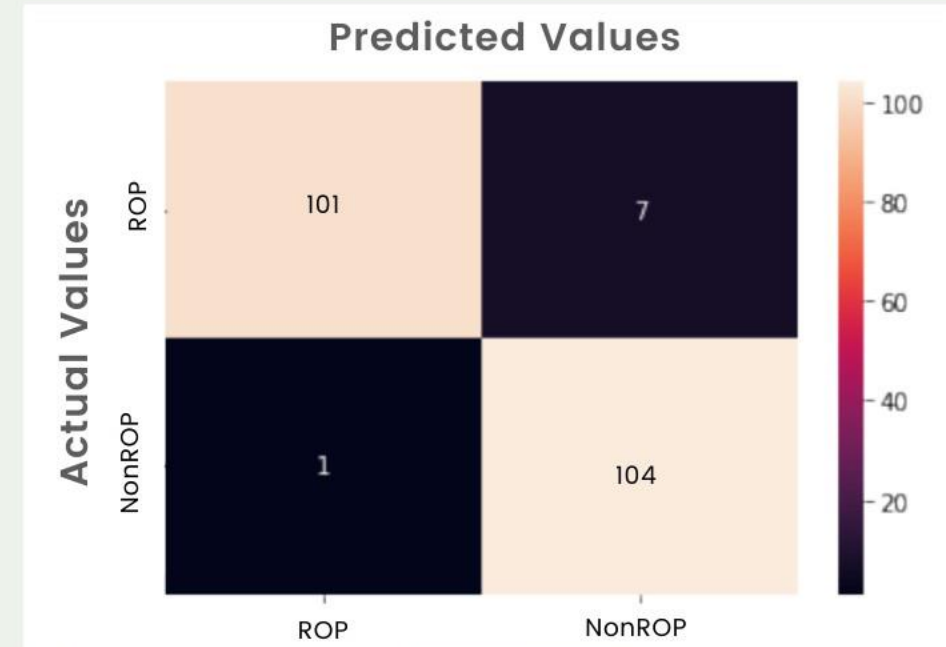
Model **Tanpa** Augmentasi



Keterangan :

Data aktual ROP dan hasil prediksi ROP : 7
Data aktual ROP dan hasil prediksi NonROP : 3
Data aktual NonROP dan hasil prediksi ROP : 0
Data aktual NonROP dan hasil prediksi NonROP : 0

Model **Dengan** Augmentasi



Keterangan :

Data aktual ROP dan hasil prediksi ROP : 101
Data aktual ROP dan hasil prediksi NonROP : 1
Data aktual NonROP dan hasil prediksi ROP : 7
Data aktual NonROP dan hasil prediksi NonROP : 104

• Evaluasi Model: Classification Report



Classification Report

Classification Report:				
	precision	recall	f1-score	support
0	0.70	1.00	0.82	7
1	0.00	0.00	0.00	3
accuracy			0.70	10
macro avg	0.35	0.50	0.41	10
weighted avg	0.49	0.70	0.58	10

Model **Tanpa** Augmentasi

Classification Report:				
	precision	recall	f1-score	support
0	0.99	0.94	0.96	108
1	0.94	0.99	0.96	105
accuracy			0.96	213
macro avg	0.96	0.96	0.96	213
weighted avg	0.96	0.96	0.96	213

Model **Dengan** Augmentasi

Terbentuk model **TERBAIK** untuk mendeteksi Retinopathy of Prematurity (ROP) pada Bayi Prematur Menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-19 adalah

Model Dengan Augmentasi Sumber Data dengan parameter

- Optimizer : Adam
- Learning rate : 2×10^{-5}
- Loss function : Sparse Categorical Cross-Entropy
- Batch size : 32
- Epoch : 20

KESIMPULAN

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 model menggunakan metode CNN dengan arsitektur VGG19. Dari training model tersebut terbentuk suatu model yang kemudian dapat digunakan untuk mengklasifikasikan data mata baru ke dalam kelas yang sudah ditentukan yaitu ROP atau non-ROP.

KESIMPULAN

2. Didapatkan analisis kinerja model dari metode Convolutional Neural Network dalam mendeteksi Retinopathy of prematurity (ROP) pada bayi prematur adalah sebagai berikut:

Model **Tanpa** Augmentasi

Akurasi	60%
Precision	30%
Recall	50%
F1-Score	37%
Loss	0.67

Model **Dengan** Augmentasi

Akurasi	96%
Precision	96%
Recall	96%
F1-Score	96%
Loss	0.14

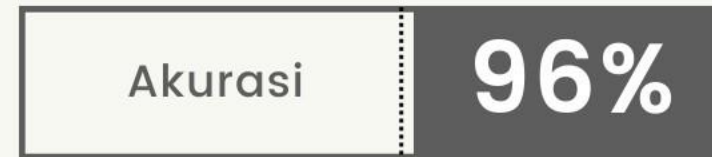
KESIMPULAN

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.

Model **Tanpa** Augmentasi



Model **Dengan** Augmentasi



Artinya, model dengan data augmentasi sangat berpengaruh dalam menghasilkan model dengan evaluasi yang lebih baik.

Referensi

- [1] 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.
- [2] 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].
- [3] 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].
- [4] S. Ollivier, "Rop_2classclassification," Kaggle, 22-Apr-2021. [Online]. Available: <https://www.kaggle.com/code/solennollivier/rop-2classclassification/data>.

TERIMA KASIH

RETINOPATHY OF PREMATURITY