

**TESIS**

**ANALISIS KOMBINASI CONVOLUTIONAL NEURAL NETWORK  
(CNN) DAN SUPPORT VECTOR MACHINE (SVM) DALAM DETEKSI  
OTOMATIS NON-PROLIFERATIVE DIABETIC RETINOPATHY**



Disusun oleh:

**Nama : Tinuk Agustin**

**NIM : 19.51.1183**

**Konsentrasi : Informatics Technopreneurship**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2021**

**TESIS**

**ANALISIS KOMBINASI CONVOLUTIONAL NEURAL NETWORK  
(CNN) DAN SUPPORT VECTOR MACHINE (SVM) DALAM DETEKSI  
OTOMATIS NON-PROLIFERATIVE DIABETIC RETINOPATHY**

**COMBINATION ANALYSIS OF CONVOLUTIONAL NEURAL NETWORK  
(CNN) AND SUPPORT VECTOR MACHINE (SVM) METHOD TO AUTO  
DETECTION NON-PROLIFERATIVE DIABETIC RETINOPATHY**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

**Nama : Tinuk Agustln**  
**NIM : 19.51.1183**  
**Konsentrasi : Informatics Technopreneurship**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2021**

**HALAMAN PENGESAHAN**

**ANALISIS KOMBINASI CONVOLUTIONAL NEURAL NETWORK (CNN)  
DAN SUPPORT VECTOR MACHINE (SVM) DALAM DETEKSI  
OTOMATIS NON-PROLIFERATIVE DIABETIC RETINOPATHY**

**COMBINATION ANALYSIS OF CONVOLUTIONAL NEURAL NETWORK (CNN)  
AND SUPPORT VECTOR MACHINE (SVM) METHOD TO AUTO DETECTION  
NON-PROLIFERATIVE DIABETIC RETINOPATHY**

Dipersiapkan dan Disusun oleh

**Tinuk Agustin**

**19.51.1183**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Rabu, 3 Februari 2021

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Februari 2021

**Rektor**

**Prof. Dr. M. Suyanto, M.M.**  
**NIK. 190302001**

**HALAMAN PERSETUJUAN**

**ANALISIS KOMBINASI CONVOLUTIONAL NEURAL NETWORK (CNN)  
DAN SUPPORT VECTOR MACHINE (SVM) DALAM DETEKSI  
OTOMATIS NON-PROLIFERATIVE DIABETIC RETINOPATHY**

**COMBINATION ANALYSIS OF CONVOLUTIONAL NEURAL NETWORK (CNN)  
AND SUPPORT VECTOR MACHINE (SVM) METHOD TO AUTO DETECTION  
NON-PROLIFERATIVE DIABETIC RETINOPATHY**

Dipersiapkan dan Disusun oleh

**Tinuk Agustin**

**19.51.1183**

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Rabu, 3 Februari 2021

**Pembimbing Utama**

**Anggota Tim Penguji**

**Prof. Dr. Ema Utami, S.Si., M.Kom.**  
NIK. 190302037

**Dr. Andi Sunvoto, M.Kom.**  
NIK. 190302052

**Pembimbing Pendamping**

**Alva Hendi Muhammad, S.T., M.Eng., Ph.D.**  
NIK. 190302493

**Hanif Al Fatta, M.Kom.**  
NIK. 190302096

**Prof. Dr. Ema Utami, S.Si., M.Kom.**  
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 Februari 2021  
**Direktur Program Pascasarjana**

**Dr. Kusriani, M.Kom.**  
NIK. 190302106

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : **Tinuk Agustin**  
NIM : **19.51.1183**  
Konsentrasi : **Informatics Technopreneurship**

Menyatakan bahwa Tesis dengan judul berikut:

**Analisis Kombinasi Convolutional Neural Network (CNN) Dan Support Vector Machine (SVM) Dalam Deteksi Otomatis Non-Proliferatif Diabetic Retinopathy**

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom.  
Dosen Pembimbing Pendamping : Hanif Al Fatta, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 3 Februari 2021

Yang Menyatakan,

  
Tinuk Agustin

## HALAMAN PERSEMBAHAN

Dengan segala puji dan syukur kepada Allah Subhanahu wa Ta'ala dan atas dukungan serta doa dari orang-orang tercinta, akhirnya tesis ini dapat terselesaikan dengan baik. Oleh karena itu, dengan rasa bangga dan bahagia penulis persembahkan tesis ini kepada:

1. Suami penulis, M. Luthfi Rahmadi yang telah memberikan dukungan moril maupun materi serta doa yang tiada henti untuk kesuksesan penulis.
2. Orang tua penulis, Bpk. Mahmud Arifin dan Ibu Kasmiatun yang telah merawat, mendidik, mensupport, dan selalu mendoakan atas kebahagiaan dan kesuksesan penulis.
3. Mertua penulis, Bpk. M. Hari P. dan Ibu Purwatiningsih yang selalu mendukung dan mendoakan penulis.
4. M. Yahya Rahmadi, Afifah N.A. dan Agung T.P.A. yang selalu menemani, menyemangati dan membantu penulis.
5. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. dan bapak Hanif Al Fatta, M.Kom. selaku dosen pembimbing dan juga Bpk. Dr. Andi Sunyoto, M.Kom., dan Bpk. Alva Hendi Muhammad, S.T.M.Eng.Ph.D. selaku dosen penguji penulis yang telah memberikan arahan serta saran sehingga tesis ini dapat terselesaikan dengan baik.
6. Keluarga besar kelas 19-S2TI-A, terima kasih atas bantuan dan kebersamaan yang sangat berarti bagi penulis.

Terima kasih atas semangat dan dukungan dari semua pihak. Semoga tesis ini dapat bermanfaat dan berguna di masa yang akan datang. Aamiin.

#### HALAMAN MOTTO

*"Don't be sad, Allah is with us"*

Janganlah bersedih, Allah bersama kita

*"Life is just about being grateful and patient"*

Hidup hanyalah tentang bagaimana bersyukur dan bersabar

*"If you really want something, gradually you will soon find a way"*

Jika kamu benar-benar menginginkan sesuatu, lambat laun kamu pasti akan segera menemukan caranya

*"Every difficulty always has convenience. Every problem has a solution"*

Setiap kesulitan selalu ada kemudahan. Setiap masalah pasti ada solusi

*"Stop underestimating yourself"*

Berhenti meremehkan diri kamu sendiri

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhanallau wata'ala yang telah memberikan kesehatan jasmani dan rohani serta petunjuk dan kekuatan kepada penulis sehingga tesis yang berjudul "Analisis Kombinasi Convolutional Neural Network dan Support Vector Machine dalam Deteksi Otomatis Non-Proliferative Diabetic Retinopathy" dapat terselesaikan dengan baik. Kritik dan saran sangat diharapkan penulis agar dapat lebih baik lagi di kemudian hari. Dalam penyusunan dan penulisan tesis ini tidak terlepas dari bantuan dan bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis dengan senang hati menyampaikan terima kasih kepada:

1. Prof. Dr. M. Suyanto, M.M. selaku rektor Universitas AMIKOM Yogyakarta.
2. Dr. Kusri, M.Kom. selaku Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta yang telah menunjuk dosen pembimbing sehingga memperlancar penulis dalam menyelesaikan tesis.
3. Prof. Dr. Ema Utami, S.Si., M.Kom. dan Hanif Al Fatta, M.Kom. selaku pembimbing utama dan pendamping yang telah mencurahkan perhatian, bimbingan, nasihat, doa, dan kepercayaan yang sangat berarti bagi penulis serta telah meluangkan waktu dalam memberikan arahan dan masukan selama penelitian dan penyusunan tesis.



4. Andi Sunyoto, M.Kom. dan Alva Hendi Muhammad, ST., M.Eng. selaku dosen penguji yang telah memberikan saran dan masukan yang membangun.
5. Suami, orang tua, mertua, adik-adik dan anak yang telah memberikan doa dan motivasi sehingga menjadi penyemangat bagi penulis dalam mengerjakan tesis.
6. Teman-teman yang telah memberikan semangat dan dukungan sehingga tesis ini dapat terselesaikan dengan baik.
7. Semua pihak yang telah membantu, baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu per satu.

Akhir kata, semoga tesis ini bermanfaat, khususnya bagi penulis dan umumnya bagi masyarakat dalam rangka menambah wawasan pengetahuan.

Yogyakarta, 3 Februari 2021

Penulis

## DAFTAR ISI

|   |       |
|---|-------|
| HALAMAN JUDUL.....                      | ii    |
| HALAMAN PENGESAHAN.....                 | iii   |
| HALAMAN PERSETUJUAN.....                | iv    |
| HALAMAN PERNYATAAN KEASLIAN TESIS ..... | v     |
| HALAMAN PERSEMBAHAN .....               | vi    |
| HALAMAN MOTTO .....                     | vii   |
| KATA PENGANTAR.....                     | viii  |
| DAFTAR ISI.....                         | x     |
| DAFTAR TABEL.....                       | xv    |
| DAFTAR GAMBAR.....                      | xvi   |
| INTISARI.....                           | xviii |
| <i>ABSTRACT</i> .....                   | xix   |
| BAB I PENDAHULUAN.....                  | 1     |
| 1.1. Latar Belakang Masalah .....       | 1     |
| 1.2. Rumusan Masalah.....               | 5     |
| 1.3. Batasan Masalah .....              | 6     |
| 1.4. Tujuan Penelitian .....            | 6     |
| 1.5. Manfaat Penelitian .....           | 7     |
| 1.6. Hipotesa .....                     | 7     |
| BAB II TINJAUAN PUSTAKA.....            | 8     |
| 2.1. Tinjauan Pustaka.....              | 8     |

|  |    |
|--|----|
| 2.2. Keaslian Penelitian.....                                    | 14 |
| 2.3. <i>Diabetic Retinopathy</i> .....                           | 17 |
| 2.3.1 <i>Non-Proliferative Diabetic Retinopathy (NPDR)</i> ..... | 17 |
| 2.3.2 <i>Proliferative Diabetic Retinopathy (PDR)</i> .....      | 18 |
| 2.4. Preprocessing.....  | 19 |
| 2.4.1 Pengelompokan Data.....                                    | 19 |
| 2.4.2 Data Augmentasi .....                                      | 21 |
| 2.5. <i>Mechine Learning</i> .....                               | 22 |
| 2.5.1 <i>Supervised Learning</i> .....                           | 22 |
| 2.5.2 <i>Unsupervised Learning</i> .....                         | 23 |
| 2.5.3 <i>Support Vector Machine (SVM)</i> .....                  | 23 |
| 2.6. <i>Deep learning</i> .....                                  | 25 |
| 2.7. <i>Convolutional Neural Network (CNN)</i> .....             | 26 |
| 2.7.1 <i>Convolution Layer</i> .....                             | 27 |
| 2.7.2 <i>RELU</i> .....  | 28 |
| 2.7.3 <i>Pooling Layer</i> .....                                 | 28 |
| 2.7.4 <i>Global Average Pooling</i> .....                        | 28 |
| 2.7.5 <i>Fully Layer</i> .....                                   | 29 |
| 2.8. <i>Model Pre-Trained</i> .....                              | 29 |
| 2.9. <i>Transfer learning</i> .....                              | 29 |
| 2.9.1 <i>Inception V3</i> .....                                  | 30 |

|   |           |
|---|-----------|
| 2.9.2 VGG16 .....                                   | 31        |
| 2.9.3 Resnet50 .....                                | 31        |
| 2.10. <i>Fine-tuning</i> .....                      | 32        |
| 2.11. <i>Multi Class Confusion Matrix</i> .....     | 32        |
| <b>BAB III METODE PENELITIAN</b> .....              | <b>35</b> |
| 3.1. Jenis, Sifat, dan Pendekatan Penelitian.....   | 35        |
| 3.2. Metode Pengumpulan Data.....                   | 35        |
| 3.3. Metode Analisis Data.....                      | 36        |
| 3.4. Alur Penelitian.....                           | 36        |
| <b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN</b> ..... | <b>41</b> |
| 4.1. Tahap Membangun Dataset.....                   | 41        |
| 4.1.1 Pengumpulan Data.....                         | 41        |
| 4.1.2 Pengelompokan Data.....                       | 41        |
| 4.1.3 Perlakuan Terhadap Data .....                 | 42        |
| 4.2. Tahap Proses Analisis.....                     | 42        |
| 4.3. Tahap Implementasi Praproses .....             | 43        |
| 4.3.1 Implementasi Fungsi Load Data.....            | 44        |
| 4.3.2 Implementasi <i>Preprocessing</i> .....       | 44        |
| 4.3.3 Implementasi Data Augmentasi .....            | 45        |
| 4.4. Tahap membangun Model .....                    | 46        |
| 4.5. Tahap Implementasi Skenario Pertama .....      | 48        |
| 4.5.1 Implementasi Transfer Learning .....          | 48        |

|        |   |    |
|--------|---|----|
| 4.5.2  | Fine-Tuning VGG16 .....   | 49 |
| 4.5.3  | Implementasi Model Pretrained .....   | 53 |
| 4.6.   | Tahap Implementasi Skenario Kedua .....   | 54 |
| 4.7.   | Tahap Implementasi Skenario Ketiga .....  | 55 |
| 4.8.   | Tahap Implementasi Skenario Keempat .....   | 56 |
| 4.9.   | Tahap Prediksi Klasifikasi Menggunakan <i>Confusion Matrix</i> .....  | 57 |
| 4.10.  | Analisis dan Pembahasan .....   | 58 |
| 4.10.1 | Konfigurasi Penelitian .....  | 58 |
| 4.10.2 | Hasil Implementasi Preprocessing .....  | 59 |
| 4.10.3 | Hasil Implementasi Data Augmentasi .....  | 59 |
| 4.10.4 | Hasil Implementasi Arsitektur Transfer Learning pada Klasifikasi menggunakan metode CNN .....               | 61 |
| 4.10.5 | Hasil Implementasi Arsitektur Transfer Learning pada Klasifikasi Menggunakan Metode Kombinasi CNN SVM ..... | 62 |
| 4.10.6 | Hasil Implementasi Model Fine Tuning VGG16 .....  | 64 |
| 4.10.7 | Hasil Implementasi Klasifikasi menggunakan <i>Model Pretrained</i> CNN .....                                | 64 |
| 4.10.8 | Perbandingan Rasio Data ( <i>Data Split</i> ) .....   | 65 |
| 4.10.9 | Hasil Implementasi Optimalisasi Model Kombinasi CNN SVM .....   | 70 |
| 4.10.1 | Perbandingan Hasil Klasifikasi CNN dan kombinasi CNN dan SVM .....  | 70 |

|                |  |    |
|----------------|--|----|
| 4.10.10        | Grafik Perbandingan Hasil Klasifikasi CNN dan kombinasi CNN dan SVM                                      | 71 |
| 4.10.11        | Grafik Perbandingan Hasil Klasifikasi Peningkatan Akurasi yang Diperoleh dari Evaluasi Awal hingga Akhir | 72 |
| 4.10.12        | Hasil perhitungan <i>Confusion Matrix</i>  | 74 |
| 4.11.          | Hasil Analisis Penelitian Terkait  | 76 |
| BAB V PENUTUP  |  | 78 |
| 5.1.           | Kesimpulan   | 78 |
| 5.2.           | Saran  | 78 |
| DAFTAR PUSTAKA |  | 80 |



## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 2. 1. Matriks Literatur Review dan Posisi Penelitian.....                                      | 14 |
| Tabel 2. 2. <i>Multi Class Confusion Matrix</i> .....  | 33 |
| Tabel 4. 1. Pembagian Data .....   | 41 |
| Tabel 4. 2. CNN <i>Train</i> Parameter.....  | 47 |
| Tabel 4. 3. arsitektur mode fine tuning VGG16.....   | 50 |
| Tabel 4. 4. Parameter Fine Tuning VGG16.....   | 50 |
| Tabel 4. 5. Parameter Pelatihan menggunakan CNN.....   | 58 |
| Tabel 4. 6. Hasil Implementasi <i>Transfer Learning</i> Klasifikasi pada CNN <i>Softmax</i><br>..... | 61 |
| Tabel 4. 7. Hasil Perbandingan Model Kombinasi CNN SVM.....  | 63 |
| Tabel 4. 8. Hasil <i>Fine Tuning</i> .....   | 64 |
| Tabel 4. 9. Tabel Perbandingan <i>Pretrained Model</i> .....   | 65 |
| Tabel 4. 10. Hasil Perbandingan Rasio Data.....  | 66 |
| Tabel 4. 11. Hasil Perbandingan Model Kombinasi optimalisasi CNN SVM.....                            | 70 |
| Tabel 4. 12. Perbandingan Hasil Model CNN <i>Softmax</i> dan CNN SVM.....                            | 71 |
| Tabel 4. 13. Grafik Confusion Matrix.....  | 74 |
| Tabel 4. 14. Perhitungan Hasil Confusion Matrix .....  | 75 |
| Tabel 4. 15. Perbandingan Penelitian Terkait .....   | 76 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2. 1. Ilustrasi Perbedaan Retina Normal dan <i>Retinopathy</i> .....                                  | 17 |
| Gambar 2. 2. Perbandingan retina NPDR .....  | 18 |
| Gambar 2. 3. Penggunaan Tiga Set Rasio data Split .....  | 20 |
| Gambar 2. 4. Hyperplane yang memisahkan dua kelas positif (+1) dan negatif (-1)<br>.....                     | 24 |
| Gambar 2. 5. Ilustrasi Pengaruh Parameter <i>C</i> (Ben-Hur & Weston, 2010).....                             | 25 |
| Gambar 2. 6. Cara kerja CNN.....   | 27 |
| Gambar 2. 7 <i>Pre-Trained VGG16</i> dan <i>Fine Tuning</i> .....  | 32 |
| Gambar 3. 1. Alur Penelitian.....  | 38 |
| Gambar 4. 1. Arsitektur Model Klasifikasi yang Dibangun .....  | 47 |
| Gambar 4. 2. Hasil <i>Green Channel</i> dan CLAHE.....   | 59 |
| Gambar 4. 3. Hasil dari Augmentasi <i>Horizontal Flip</i> .....  | 60 |
| Gambar 4. 4. Hasil dari Augmentasi <i>Random Brightness</i> .....  | 60 |
| Gambar 4. 5. Hasil dari Augmentasi <i>Chanel Shift Range</i> .....   | 61 |
| Gambar 4. 6. Ekstraksi Fitur dari VGG16 .....  | 63 |
| Gambar 4. 7. Ekstraksi Fitur Resne50 .....   | 63 |
| Gambar 4. 8. Ekstraksi Fitur InceptionV3 .....   | 63 |
| Gambar 4. 9. Grafik Fit Model Perbandingan 2 Split Data Dengan 3 Split Data<br>Pada Model VGG 16 .....       | 67 |
| Gambar 4. 10. Grafik Fit Model Perbandingan 2 Split Data Dengan 3 Split Data<br>Pada Model Inceptionv3 ..... | 68 |



|  |    |
|--|----|
| Gambar 4. 11. Grafik Fit Model Perbandingan 2 Split Data Dengan 3 Split Data<br>Pada Model Resnet 50 ..... | 69 |
| Gambar 4. 12. Perbandingan Klasifikasi Metode CNN dengan Metode Kombinasi<br>CNN dan SVM .....             | 71 |
| Gambar 4. 13 Peningkatan akurasi pada model VGG16 .....  | 73 |
| Gambar 4. 14. Peningkatan akurasi pada model Resnet50 .....  | 73 |
| Gambar 4. 15. Peningkatan akurasi pada model InceptionV3 .....   | 74 |



## INTISARI

Dalam penelitian Analisis kombinasi *Convolutional Neural Network* dan *Support Vector Machine* untuk deteksi otomatis *Non-Proliferative Diabetic Retinopathy* ini bertujuan untuk mengklasifikasikan tingkat keparahan NPDR (ringan, sedang, berat) dengan menggunakan metode CNN sebagai metode ekstraksi fitur dan kemudian akan diklasifikasikan dengan menggunakan metode SVM. Dalam penelitian ini juga akan membandingkan performa dari metode CNN softmax dan metode kombinasi CNN SVM yang akan dinialisis berdasarkan nilai akurasi dan sensitivitas.

Dataset dalam penelitian ini terbagi menjadi data latih, data validasi, dan data uji dengan perbandingan 70:15:15. Dan menggunakan teknik augmentasi *horizontal flip* untuk memperbanyak data, selain itu juga menggunakan data augmentasi untuk menambah variasi data pada saat pelatihan. Sebelum memasuki ekstraksi fitur dataset di preproses dengan hanya mengambil chanel hijau dari RGB gambar yang kemudian ditingkatkan kontrasnya dengan menggunakan CLAHE. Data input hasil preproses ini kemudian di ekstraksi fitur menggunakan CNN dengan model pretrained dari Resnet50, VGG16, dan InceptionV3. Setelah di latih dan mendapatkan model terbaik kemudian model disimpan untuk digunakan pada proses klasifikasi menggunakan SVM.

Penelitian ini menyimpulkan bahwa metode kombinasi CNN SVM lebih unggul dibandingkan metode klasifikasi menggunakan CNN softmax. Selain itu, hal yang mempengaruhi adalah penggunaan arsitektur CNN yang digunakan untuk ekstraksi fitur. Penelitian ini menyimpulkan bahwa model arsitektur dengan menggunakan pretrained dari Resnet50 kombinasi SVM memberikan hasil akurasi yang lebih tinggi dibandingkan model lainnya. Dengan hasil akurasi yang diperoleh 99,77% pada pretrained Resnet50 kombinasi SVM.

Kata kunci: Convolutional Neural Network, Support Vector Machine, Diabetic Retinopathy, transfer learning, klasifikasi multikelas

## **ABSTRACT**

*In this research, the combination analysis of Convolutional Neural Network and Support Vector Machine for automatic detection of Non-Proliferative Diabetic Retinopathy aims to classify the severity of NPDR (mild, moderate, severe) using the CNN method as a feature extraction method and will then be classified using the SVM method. In this study, we will also compare the performance of the CNN softmax method and the CNN SVM combination method which will be analyzed based on the accuracy and sensitivity values.*

*The dataset in this study is divided into training data, validation data, and test data with a ratio of 70:15:15. And using the horizontal flip augmentation technique to multiply the data, besides that, it also uses augmentation data to increase data variation during training. Before entering feature extraction, the dataset is preprocessed by only taking the green channel from the RGB image which then increases the contrast using CLAHE. The preprocessed input data was then feature extraction using CNN with pretrained models from Resnet50, VGG16, and InceptionV3. After being trained and getting the best model, the model is stored for use in the classification process using SVM.*

*This study concluded that the combined CNN SVM method was superior to the CNN softmax classification method. In addition, the thing that affects is the use of the CNN architecture used for feature extraction. This study concludes that an architectural model using the pretrained Resnet50 SVM combination provides higher accuracy results than other models. With the results obtained accuracy 99.77% on the pretrained Resnet50 combination of SVM.*

*Keyword: Convolutional Neural Network, Support Vector Machine, Diabetic Retinopathy, transfer learning, multiclass classification*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Retinopati diabetik atau juga dikenal dengan *diabetic retinopathy* (DR) adalah komplikasi diabetes melitus yang menyerang retina mata yang disebabkan oleh kadar gula yang tinggi sehingga menimbulkan kerusakan pada pembuluh darah retina. Hal ini menyebabkan pembuluh darah membengkak dan mengalami kebocoran yang mengakibatkan terjadinya penumpukan cairan (*exudates*) yang bercampur lemak serta pendarahan pada retina. *International Diabetes Federation* (IDF) mencatat pada 2019 diperkirakan 463 juta penderita *diabetes mellitus* di seluruh dunia dan akan semakin meningkat. Sekitar sepertiganya mengalami tanda-tanda DR dengan sepertiganya merupakan DR yang mengancam penglihatan (Melo et al., 2018). Menurut *The Early Treatment of Diabetic Retinopathy Study* (ETDRS), tahap awal penyakit mata diabetes dikenal dengan sebutan *Non-Proliferative Diabetic Retinopathy* (NPDR). Tahap awal DR ini terbagi menjadi *mild*, *moderate* dan *severe* NPDR, dan pada tahapan yang lebih berat *Proliferative Diabetic Retinopathy* (PDR) mengarah pada pelepasan retina (Christodoulidis et al., 2016). Secara objektif, gejala yang ditunjukkan pada penderita DR antara lain: fase awal ditandai dengan adanya *microaneurysms* (MA), *hemorrhages* (HE); *hard exudate* (EX); *soft exudates* (SE) (Fu et al., 2016). DR adalah penyebab utama kebutaan pada pasien diabetes yang seharusnya dapat dicegah. Deteksi dini dan

penanganan pada fase awal DR sangat penting agar dapat mencegah terjadinya perburukan fungsi penglihatan pasien (Abramoff et al., 2010).

Namun pada sebuah penelitian terhadap dokter klinik yang sudah terlatih untuk membaca hasil foto funduskopi menunjukkan bahwa dokter klinik tidak dapat mendeteksi dini adanya kelainan karena mayoritas dokter baru bisa mendeteksi DR setelah terjadinya *exudates* atau pendarahan sehingga tidak dapat secepatnya merujuk pasien ke spesialis mata sub retina (F et al., 2008). Karena tahap awal penyakit DR tidak memiliki gejala nyata yang dapat dirasakan oleh pasien, deteksi dini hanya dapat dilakukan dengan pengambilan citra fundus retina. Gejala awal yang terlihat pada citra fundus adalah dengan munculnya *microaneurysms* yang terlihat seperti titik merah kecil. Karena ukurannya yang kecil dan mirip dengan daerah di sekitarnya ditambah lagi dengan faktor resolusi dan kontras dari citra membuat diagnosis menjadi rumit dan membutuhkan waktu yang lama (Indumathi & Sathananthavathi, 2019). Deteksi gejala yang terkait dengan DR seperti yang disebutkan di atas adalah sumber daya yang sangat bermanfaat dalam pengembangan analisis citra retina dan algoritma pencarian untuk deteksi dini DR (Porwal et al., 2018).

Oleh sebab itu untuk mendapatkan diagnosis yang tepat dan tindak lanjut yang cepat, deteksi awal DR berbasis teknologi perlu dikembangkan (Ting et al., 2016). Peran kemajuan teknologi saat ini tentang sistem komputasi, pembelajaran mesin, dan pembelajaran mendalam adalah peluang besar bagi para peneliti untuk terus mengembangkan penelitian dalam bidang kesehatan (Shortliffe & Blois, 2001). Teknologi berbantuan komputer dengan menggunakan *artificial intelligence*

(AI) efektif dalam membantu *screening* DR pada tahap awal. AI dapat mengurangi beban kerja dalam membaca hasil foto funduskopi, memperoleh hasil yang lebih cepat dan akurat serta lebih murah daripada cara manual (Fleming et al., 2010). *Mechine learning* dan *deep learning* adalah bagian dari AI yang sangat populer. Karena kemampuannya untuk mencerna rangkaian data, mengambil kesimpulan berdasarkan data yang dianalisis, dan dapat menggunakan kesimpulan tersebut untuk menyelesaikan tugasnya dengan cara paling efektif. *Deep learning* yang dipopulerkan oleh Geoffrey Hinton pada tahun 2006 mengalami perkembangan pesat setelah ditemukannya GPU oleh Andrew et al. di 2009. Teknik *deep learning* ini sangat sesuai dalam analisis citra medis. Metode *deep learning* yang banyak digunakan untuk melakukan klasifikasi tahapan DR adalah dengan menggunakan metode CNN. Karena CNN memiliki kemampuan ekstraksi fitur yang sangat baik terutama dalam pengenalan gambar (Y. D. Zhang et al., 2019).

Kelebihan utama metode CNN dalam pengenalan gambar adalah dari arsitekturnya yang mampu mengenali informasi prediktif dari suatu objek. Namun metode *deep learning* juga mempunyai kekurangan yaitu tidak mampu mengolah data dalam jumlah kecil secara maksimal. Selain itu *Neural Network* juga sering mengalami *overfitting* karena *overtrained* (Santosa, 2007). Sedangkan pada SVM istilah *overfitting* tidak dikenal karena *training* hanya perlu dilakukan sekali saja dan akan mendapatkan hasil optimal.

Secara umum arsitektur CNN dibagi menjadi tiga *layer* dengan kontribusi CNN di *feature extraction layer* adalah pada *convolution* dan *pooling layer*. Dan *fully-connected layer* yang digunakan untuk melakukan klasifikasi hanya dapat

digunakan pada akhir jaringan. Karena untuk masuk ke sebuah *fully-connected layer* data dari *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sehingga menyebabkan data kehilangan informasi spasialnya (Stanford, 2020). Oleh sebab itu penggunaan *layer* ini pada jaringan CNN tidak banyak dipakai. Menyiasati hal tersebut (Qomariah et al., 2019) melakukan penelitian dengan menggabungkan dua metode klasifikasi CNN dan SVM untuk membedakan retina normal dengan retina DR parah dan memberikan hasil yang cukup baik dalam kasifikasi DR. Begitu pula yang dilakukan (Khojasteh et al., 2019) untuk deteksi eksudat menggunakan klasifikasi dengan mengkombinasi CNN dan SVM. Namun kedua penelitian ini mengklasifikan objek menjadi dua kelas, apalagi SVM sangat terkenal baik dalam klasifikasi dua kelas. Namun demikian SVM juga dapat digunakan dengan baik pada klasifikasi multikelas, seperti pekerjaan yang telah dilakukan oleh (Gong et al., 2019) yang menggunakan kombinasi CNN dan SVM untuk diagnosis kesalahan cerdas dari mesin berputar yang diklasifikasikan menjadi 10 kelas. Dan mendapatkan akurasi sebesar 99,12% untuk klasifikasi menggunakan CNN softmax dan 99,94% untuk klasifikasi menggunakan metode kombinasi CNN SVM.

Namun hasil tersebut berbeda dengan peneltiam yang telah dilakukan oleh (Araujo et al., 2017), dengan menggunakan kombinasi CNN SVM untuk mendeteksi kanker payudara dan menunjukkan bahwa klasifikasi menggunakan CNN lebih unggul daripada menggunakan metode kombinasi CNN SVM pada empat kelas klasifikasi. Arsitektur CNN yang dibangun oleh Araujo adalah model yang beliau costum sendiri. Lalu bagaimana jika menggunakan arsitektur yang

sudah terbukti handal dan pernah memenangkan berbagai penghargaan. Disini (Y. Wang et al., 2018) telah menguji metode kombinasi CNN SVM untuk klasifikasi kanker payudara menjadi empat kelas klasifikasi dengan menggunakan model transfer learning VGG16 dan mendapatkan akurasi sebesar 91,7 % untuk metode CNN *softmax* dan 80,6 % untuk metode kombinasi CNN SVM.

Berdasarkan latar belakang tersebut, apakah akurasi kombinasi CNN SVM untuk klasifikasi NPDR memberikan akurasi sebaik yang dilakukan oleh penelitian Gong ataukah justru metode klasifikasi CNN *softmax* sendirilah yang memberikan akurasi yang lebih baik seperti pada penelitian yang telah dilakukan oleh Araujo dan Wang. Penelitian ini bermaksud untuk mengklasifikasikan level keparahan NPDR (*mild, moderate, severe*) dengan menggunakan metode CNN sebagai metode ekstraksi fitur dan kemudian akan diklasifikasikan dengan menggunakan metode SVM. Dalam penelitian ini juga akan membandingkan performa dari metode CNN *softmax* dan kombinasi CNN SVM yang akan dianalisis berdasarkan nilai akurasi dan sensitivitas, dengan menggunakan model CNN dari pembelajaran transfer VGG16, Resnet50, dan InceptionV3. Metode terbaik digunakan untuk mencari nilai akurasi, sensitivitas dan spesifitas.

## 1.2. Rumusan Masalah

Bedasarkan latar belakang yang telah diuraikan dalam penelitian ini dirumuskan masalah – masalah sebagai berikut :

- a. Metode klasifikasi manakah diantara metode CNN *softmax* dan metode kombinasi CNN SVM yang dapat memberikan nilai akurasi tinggi pada klasifikasi NPDR?



- b. Dengan menggunakan model arsitektur CNN manakah yang dapat memberikan nilai akurasi yang terbaik pada klasifikasi NPDR?
- c. Berapakah nilai akurasi, sensitivitas dan spesifitas tertinggi yang diperoleh dari teknik klasifikasi terbaik yang telah di uji?

### 1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

- a. Penelitian ini melakukan klasifikasi gambar DR menggunakan 984 citra fundus retina dengan format .tiff yang telah di evaluasi dan diberi label oleh *ophthalmologists* menjadi 4 label yaitu, *normal* pada level 0, *mild* pada level 1, *moderate* pada level 2, dan *severe* pada level 3.
- b. Data merupakan data sekunder yang diperoleh dari *messidor database* (Decenciére et al., 2014).
- c. Untuk ekstraksi fitur dalam penelitian ini menggunakan arsitektur pembelajaran transfer VGG16, Resnet50 dan InceptionNet V3
- d. Penelitian ini akan membahas perbandingan metode klasifikasi menggunakan pembelajaran mendalam (CNN) dan pembelajaran *transfer* (SVM)
- e. Pengujian setiap skenario dilakukan berdasarkan nilai akurasi dan sensitivitas

### 1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

- a. Mengetahui perbandingan performa akurasi metode klasifikasi CNN *softmax* dan metode kombinasi CNN SVM
- b. Mengetahui model arsitektur CNN yang bekerja baik dengan memberikan nilai akurasi tertinggi.

- c. Mengetahui nilai rata-rata akurasi, sensitivitas dan spesifitas berdasarkan metode terbaik yang telah dibandingkan dalam klasifikasi keparahan NPDR.

### 1.5. Manfaat Penelitian

Adapun manfaat yang akan didapat dari penelitian ini sebagai berikut:

- a. Model yang diterapkan mampu dijalankan dan menghasilkan akurasi, sensitivitas dan spesifitas yang tinggi.
- b. Mendapatkan metode optimisasi kombinasi CNN SVM yang dapat meningkatkan nilai akurasi, sensitivitas, dan spesifitas lebih baik daripada klasifikasi menggunakan CNN *softmax*.
- c. Dapat diimplementasikan pada sistem *screening otomatis* deteksi DR. Kami berharap dengan memberikan diagnosa gejala DR yang akurat dan cepat dapat menyelamatkan penglihatan pasien.
- d. Dapat dijadikan referensi untuk penelitian lain dalam mengkaji penelitian yang sama atau bisa menjadikan wawasan untuk pengembangan penelitian selanjutnya.

### 1.6. Hipotesa

Secara teori SVM sangat baik digunakan pada klasifikasi dua kelas namun SVM juga dapat digunakan untuk klasifikasi multi kelas. Berdasarkan penelitian sebelumnya tentang klasifikasi menggunakan kombinasi CNN SVM memberikan kesimpulan yang beragam. Metode kombinasi CNN SVM pada klasifikasi multi kelas NPDR dapat memberikan hasil akurasi yang melampaui klasifikasi dengan menggunakan CNN *Softmax*. Arsitektur CNN yang digunakan juga akan mempengaruhi hasil akurasi.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Penelitian ini dibuat dengan terlebih dahulu melakukan studi kepustakaan dari penelitian – penelitian dan sumber – sumber lainnya yang nantinya akan dijadikan sebagai tinjauan, masukan, atau perbandingan. Dari studi kepustakaan yang dilakukan, penulis menemukan beberapa penelitian yang mendorong untuk mengangkat tema seperti diatas. Penelitian tersebut membahas tentang topik yang terkait dengan penelitian penulis, antara lain penelitian mengenai metode yang akan digunakan dan penelitian yang berhubungan dengan pendeteksian dan klasifikasi tingkat keparahan DR.

Sebelumnya banyak penelitian yang melakukan tahap peningkatan kualitas gambar dengan berbagai metode. Penelitian (Soomro et al., 2016) mengusulkan Teknik peningkatan kualitas gambar berbasis operasi morfologi pada gambar fundus retina dengan menggunakan CLAHE (*Contrast Limited Adaptive Histogram Equalization*) untuk meningkatkan gambar pembuluh darah. Pada penelitian (Rakshitha et al., 2017) penulis menggunakan metode baru untuk memperbaiki citra retina dengan transformasi *wavelet*, *curvelet* dan *Conterlet* sehingga menghasilkan ekstraksi fitur dan perbandingan yang berbeda diantara ketiga transformasi tersebut.

Penelitian awal yang membahas deteksi otomatis MA adalah (Lay et al., 1983) diusulkan dengan konsep matematika *morfologi* menggunakan *fluorescein angiogram* namun metode ini mahal, lama, dan tidak cocok untuk semua orang.

Pada penelitian (Fleming et al., 2006) menggunakan berbagai metode normalisasi kontras untuk membedakan titik-titik antara MA dalam gambar fundus. Namun algoritma mereka memiliki tingkat sensitivitas dan spesifitas yang rendah sehingga tidak efisien untuk mendeteksi MA. Ada banyak penelitian terbaru tentang pendeteksian MA dengan hasil yang lebih baik dibandingkan dengan teknik sebelumnya.

(Rocha et al., 2012) mengusulkan sebuah teknik untuk mendeteksi lesi retina dan mengklasifikasikan gambar fundus berdasarkan ada atau tidaknya *Point of Interest* (PoIs) pada kamus visual kata yang dibangun. Penelitian (Lazar & Hajdu, 2013) menggunakan metode klasifikasi *Naïve Bayes* untuk menemukan satu set atribut bentuk, ukuran dan ketinggian *peak* berdasarkan analisis *cross-section profiles* dan metode *peak detection*. Penelitian (Wu et al., 2017) melanjutkan penelitian (Lazar & Hajdu, 2013) dengan memperkenalkan dua puluh tujuh karakteristik fitur dan menggunakan klasifikasi *K-Nearest Neighbors* (KNN) untuk mengklasifikasi MA.

(Rosas-Romero et al., 2015) melakukan berbagai Langkah *preprocessing* salah satunya dengan penerapan aplikasi *hit-or-miss* transmisi pada citra biner untuk mengeluarkan pembuluh darah dan menemukan kandidat MA kemudian menggunakan *Principal Component Analysis* (PCA) untuk mendeteksi MA yang sebenarnya. Walaupun demikian, keakuratan teknik ini belum memadai untuk digunakan sebagai alat diagnostik DR. (Zhou et al., 2017) mengusulkan metode klasifikasi *unsupervised* berdasarkan PCA untuk mendeteksi MA yang dapat

memecahkan masalah ketidak seimbangan kelas, namun ada beberapa *false-positive* (FP) selama ekstraksi fitur.

(S. Wang et al., 2017) mengusulkan metode menggunakan *Singular Spectrum Analysis* dan KNN untuk mendeteksi MA dan mendapatkan *false-negatives* (FN) karena ada MA yang buram dan kontras yang rendah, mereka juga kehilangan beberapa MA selama ekstraksi kandidat. (Long et al., 2020), mengusulkan algoritma pembelajaran mesin berdasarkan fitur *Directional Local Contrast* (DLC) untuk mendeteksi MA. Dengan meningkatkan segmentasi pembuluh darah berbasis *analyzing eigenvalues of Hessian matrix* untuk membedakan MA dan bukan-MA. Penelitian ini memiliki kelemahan karena ada kesalahan prediksi dimana FP menunjukkan bahwa bukan-MA diprediksi sebagai MA. Dan FN menunjukkan MA diprediksi sebagai bukan-MA. (Seoud et al., 2016) mengusulkan metode *Random Forest* (RF) untuk mendeteksi MA dan HE menggunakan fitur *dynamic shape features* tanpa perlu segmentasi lesi sebelumnya, namun lesi yang terkait pembuluh darah tidak ditemukan dan mengarah kepada FN.

(Srivastava et al., 2017) mengusulkan filter dengan ukuran *grid* berbeda untuk mendeteksi lesi merah dari gambar fundus yang dikombinasikan dengan *Multiple-Kernel Learning* (MKL) dan SVM. Hasil experiment dapat mendeteksi MA dan HE bahkan ketika lesi dekat dengan pembuluh darah. Namun memiliki kelemahan dengan ukuran *grid* yang tinggi dan sistem komputasi yang juga tinggi. Dilanjutkan dengan penelitian (Mazlan et al., 2020) untuk mendeteksi dan klasifikasi MA dengan melakukan segmentasi menggunakan *multilevel thresholding* untuk meminimalkan kesalahan deteksi, yang kemudian dilanjutkan untuk

membandingkan dua metode klasifikasi yaitu MLP dan SVM untuk menentukan adanya MA ataukah tidak. Hasil penelitian ini menunjukkan metode MLP lebih unggul dibandingkan metode SVM. Namun penelitian ini tidak dapat membedakan antara pembuluh darah (*blood vessel*) dengan MA. Penting untuk dapat membedakan MA dengan pembuluh darah agar mendapatkan deteksi yang benar.

(Jebaseeli et al., 2019) melakukan segmentasi pembuluh darah menggunakan *Tandem Pulse Coupled Neural Network* (TPCNN) model yang digunakan untuk pembuatan vektor fitur otomatis. Dan *Deep Vector Based Support Machine* (DLBSVM) diusulkan untuk klasifikasi dan ekstraksi pembuluh darah menghasilkan nilai akurasi yang tinggi namun hasil sensitivitasnya rendah, dimana sensitivitas menunjukkan kemampuan algoritma dapat mendeteksi pembuluh darah dengan benar. (Amel et al., 2012) mengusulkan sebuah algoritma yang menggabungkan morfologi matematika dan *K-Means clustering* untuk mendeteksi EX berdasarkan variabel gambar input dalam hal kecerahan dan kontras. (Medhi & Dandapat, 2016) melakukan penelitian untuk mendeteksi EX setelah pengangkatan *optic disk*, namun kekurangan penelitian ini melewatkan beberapa *exudates* yang samar dan tidak dapat membedakan EX dengan struktur lainnya terutama serat optik dan *vessel reflection*. Sedangkan (Prentašić & Lončarić, 2016) melalui penelitiannya menggunakan CNN dalam menentukan label setiap *pixel* dalam *sliding window* untuk setiap iterasi *output* dari CNN dikombinasikan dengan hasil *cakram optic* dapat mendeteksi segmen *exudates* secara akurat.

Penelitian sebelumnya yang dapat dijadikan rujukan untuk melakukan deteksi dan klasifikasi NPDR misalkan pada penelitian yang telah dilakukan untuk

mendeteksi eksudat (Fraz et al., 2019), deteksi lesi merah (Biyani & Patre, 2018) dan sistem ulasan tematik dengan berfokus pada diagnosis DR berbasis komputer (Norgaard & Grauslund, 2018). Penelitian (Srivastava et al., 2017) mengkombinasikan *Multiple-Kernel Learning* (MKL) dan *Support Vector Machine* (SVM). Metode ini digunakan untuk mencari lesi merah dari gambar fundus dan berhasil mendeteksi MA dan HE bahkan ketika lesi dekat dengan pembuluh darah. Namun menyebabkan sistem komputasi yang tinggi untuk *grid* yang lebih tinggi. Kemudian (Mazlan et al., 2020) mendeteksi dan mengklasifikasi MA dengan melakukan segmentasi menggunakan *multilevel thresholding* untuk meminimalkan kesalahan deteksi, yang kemudian dilanjutkan untuk membandingkan dua metode klasifikasi yaitu MLP dan SVM untuk menentukan adanya MA atau bukan MA. Hasil penelitian ini menunjukkan metode MLP lebih unggul dibandingkan metode SVM. Namun penelitian ini tidak dapat membedakan antara pembuluh darah dengan MA. (Mo et al., 2018) menyajikan jaringan residu berjenjang untuk pengenalan edema makula diabetik terdiri dari dua tahap yaitu segmentasi EX dan klasifikasi gambar dengan melatih jaringan yang berbeda secara terpisah.

Akhir-akhir ini CNN telah mengalami perkembangan yang signifikan dalam visi dan klasifikasi gambar ada banyak penelitian melakukan klasifikasi DR dengan metode CNN. Merujuk pada hasil penelitian (Litjens et al., 2017) CNN terbukti menjadi pendekatan yang menjanjikan dalam bidang analisis citra medis. Penelitian tentang CNN telah banyak dibahas misalkan pada penelitian (Eftekhari et al., 2019), (Akut, 2019), (Chudzik et al., 2018) namun pada penelitian ini hanya menggunakan sedikit data sebagai *dataset*. Data skala besar pada pasien heterogen diperlukan

untuk mendapatkan validasi yang sempurna. (X. Wang et al., 2018) memperkenalkan pendekatan berbasis analisis gambar untuk secara otomatis membedakan 5 tahap DR berdasarkan gambar funduskopi. Hasil percobaan ini menunjukkan keefektifan CNN dalam penentuan DR. Penelitian (Wan et al., 2018) dilakukan dengan melakukan peningkatan *image* terlebih dahulu kemudian menggunakan arsitektur CNN yang mengadopsi *Alexnet* (Krizhevsky et al., 2015), *VGGnet* (Simonyan & Zisserman, 2015), *Googlenet* (Szegedy, 2015), dan *ResNet* (He et al., 2016) untuk melakukan ekstraksi fitur dan dan klasifikasi sekaligus, namun kelemahan dari metode ini membutuhkan waktu komputasi data yang lebih lama dalam pelatihan dan pengujian.

Dalam penelitian untuk menguji nilai akurasi yang dihasilkan dari klasifikasi dengan menggunakan metode kombinasi CNN dan SVM dengan fitur ekstraksi CNN yang di *costom* sendiri seperti pada penelitian (Gong et al., 2019) yang menguji kinerja metode kombinasi CNN SVM untuk mendiagnosis kesalahan cerdas menunjukkan bahwa metode yang diusulkan lebih efektif dibandingkan metode SVM, KNN, BPNN, DNN, dan CNN tradisional. Hasil penelitian Gong menghasilkan akurasi sebesar 99,12% untuk CNN *softmax* dan 99,94% untuk klasifikasi CNN SVM pada klasifikasi multi kelas. Pada penelitian (Alzubaidi et al., 2020) menghasilkan akurasi CNN *softmax* 99,54% dan akurasi kombinasi CNN SVM 99,98% untuk tiga kelas klasifikasi. Berbeda dengan penelitian (Y. Wang et al., 2018) dan (Araujo et al., 2017) yang menghasilkan akurasi CNN *softmax* lebih unggul dari pada metode kombinasi CNN SVM untuk klasifikasi lima tahap DR.



## 2.2. Keaslian Penelitian

Tabel 2. 1. Matriks Literatur Review dan Posisi Penelitian

Analisis Kombinasi *Convolutional Neural Network (CNN)* dan *Support Vector Machine (SVM)* Dalam Deteksi Otomatis *Non-Proliferative Diabetic Retinopathy*

| No | Judul   | Peneliti, Media Publikasi, dan Tahun                               | Tujuan Penelitian  | Kesimpulan   | Saran atau Kelemahan   | Perbandingan  |
|----|---|--|--|--|--|---|
| 1  | <i>Classification of Diabetic Retinopathy and Normal Retinal Images using CNN and SVM</i> | Qomariah, et al., ICTS, 2019                                       | Klasifikasi 2 kelas DR menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur menggunakan pembelajaran transfer dari Alexnet, VGGnet, InceptionNet, GoogleNet, dan DenseNet. Menggunakan dataset Messidor dengan jumlah 77 gambar.   | Kombinasi <i>transfer learning</i> ResNet 50 SVM memberikan hasil akurasi tertinggi 95,83%. Penelitian ini menyimpulkan bahwa menggunakan kombinasi fitur yang diekstrak dari pembelajaran transfer CNN dan SVM dapat memberikan hasil yang baik untuk klasifikasi 2 kelas DR. | Dataset yang digunakan untuk penelitian hanya berjumlah 70 gambar untuk membedakan retina normal dan retina DR.                    | Penelitian kami menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur menggunakan model VGG16, Resnet50, dan InceptionV3. Serta membandingkan metode klasifikasi CNN dengan metode kombinasi CNN dan SVM untuk klasifikasi 4 kelas DR. |
| 2  | <i>Exudate Detection in Fundus Images Using Deeply-learnable Features</i>                 | Khojasteh, et al., <i>Computers in Biology and Medicine</i> , 2019 | Klasifikasi 2 kelas kanker payudara dari gambar mikroskopis menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur menggunakan pembelajaran transfer dari VGG16, dan membandingkannya dengan akurasi klasifikasi CNN dengan model VGG16. Dataset yang digunakan adalah ICIAR dengan jumlah 400 gambar. Dengan data augmentasi. | kinerja Resnet50 kombinasi SVM mendapatkan hasil klasifikasi yang lebih baik daripada metode lainnya yaitu dengan mendapatkan akurasi sebesar 98,2%, sensitivitas 99%, dan spesifitas 96%.   | Agar lebih seimbang dalam penilaian agar untuk membandingkan kinerja dari Resnet50 jika dibandingkan dengan Resnet50 kombinasi SVM | Penelitian kami menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur menggunakan model VGG16, Resnet50, dan InceptionV3. Serta membandingkan metode klasifikasi CNN dengan metode kombinasi CNN dan SVM untuk klasifikasi 4 kelas DR. |

Tabel 2. 1. Matriks Literatur Review dan Posisi Penelitian

Analisis Kombinasi *Convolutional Neural Network (CNN)* dan *Support Vector Machine (SVM)* Dalam Deteksi Otomatis *Non-Proliferative Diabetic Retinopathy (Lanjutan)*

| No | Judul  | Peneliti, Media Publikasi, dan Tahun                  | Tujuan Penelitian   | Kesimpulan   | Saran atau Kelemahan  | Perbandingan   |
|----|--|---|---|--|---|--|
| 3  | <i>Breast Cancer Microscope Image Classification Based on CNN with Image Deformation</i> | Wang, et al., Springer International Publishing, 2018 | Klasifikasi 4 kelas DR menggunakan metode CNN dengan model transfer Alexnet, VGG16 dan InceptionV3. menggunakan dataset Kaggle dengan jumlah 166 gambar dan dengan data augmentasi. | Nilai akurasi yang diperoleh sebesar 91,7 % untuk klasifikasi menggunakan metode CNN fine tuning VGG16. Dan menghasilkan akurasi sebesar 80,6% untuk klasifikasi menggunakan kombinasi metode CNN dan SVM. | Dalam penelitian ini tidak menjelaskan berapa jumlah data setelah ditambahkan data augmentasi.          | Penelitian kami menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur menggunakan model VGG16, Resnet50, dan InceptionV3. Serta membandingkan metode klasifikasi CNN dengan metode kombinasi CNN dan SVM untuk klasifikasi 4 kelas DR |
| 4  | <i>Diabetic Retinopathy Stage Classification using Convolutional Neural Networks</i>     | Wang, et al., IEEE, 2018                              | Klasifikasi 5 kelas DR menggunakan metode CNN dengan model transfer Alexnet, VGG16 dan InceptionV3. menggunakan dataset Kaggle dengan jumlah 166 gambar dan dengan data augmentasi. | Nilai akurasi rata-rata AlexNet, VGG16, dan InceptionNet V3 masing-masing adalah 37,43%, 50,03%, dan 63,23%. Hasil akurasi yang lebih tinggi telah dicapai oleh InceptionNet V3                            | Pertama, lebih banyak klasifikasi gambar berbasis CNN untuk lebih meningkatkan akurasi kategorisasi DR. | Penelitian kami menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur menggunakan model VGG16, Resnet50, dan InceptionV3. Serta membandingkan metode klasifikasi CNN dengan metode kombinasi CNN dan SVM untuk klasifikasi 4 kelas DR |
|    | <i>Recognition of diabetic retinopathy</i>   | Wu, et al., IEEE, 2019                                | Klasifikasi 5 kelas DR menggunakan metode CNN dengan model transfer Resnet50, VGG19 dan InceptionV3.  | Jaringan inception V3 menghasilkan akurasi terbaik mencapai 61%  | Nilai akurasi yang diperoleh masih terbilang kecil.   | Penelitian kami menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur   |

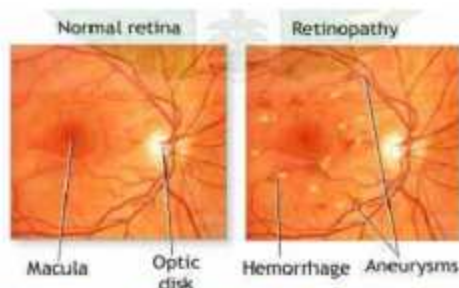
Tabel 2. 1. Matriks Literatur Review dan Posisi Penelitian

Analisis Kombinasi *Convolutional Neural Network (CNN)* dan *Support Vector Machine (SVM)* Dalam Deteksi Otomatis *Non-Proliferative Diabetic Retinopathy* (Lanjutan)

| No | Judul   | Peneliti, Media Publikasi, dan Tahun | Tujuan Penelitian  | Kesimpulan   | Saran atau Kelemahan                                  | Perbandingan  |
|----|---|--------------------------------------|--|--|---|---|
| 5  | <i>basedon transfer learning</i>  |                                      | menggunakan dataset Kaggle dengan jumlah 126 gambar dan dengan data augmentasi.  |  |   | menggunakan model VGG16, Resnet50, dan InceptionV3. Serta membandingkan metode klasifikasi CNN dengan metode kombinasi CNN dan SVM untuk klasifikasi 4 kelas DR.  |
| 6  | <i>Classification of breast cancer histology images using convolutional neural networks</i> | Araujo, et al., PLoS ONE, 2017       | Klasifikasi 4 kelas citra biopsi payudara yang diwarnai hematoksin dan eosin menggunakan metode CNN <i>custom</i> dan kombinasi CNN dan SVM. Menggunakan dataset dari Bioimaging 2015 dengan jumlah 626 gambar dan dengan data augmentasi. | Fitur yang diekstrak oleh CNN digunakan untuk melatih pengklasifikasi Support Vector Machine menghasilkan akurasi sebesar 76,9% untuk empat kelas dan 77,6% untuk hasil klasifikasi dengan menggunakan metode CNN. | Untuk dapat lebih meningkatkan akurasi yang diperoleh | Penelitian kami menggunakan metode kombinasi CNN dan SVM. Dengan ekstraksi fitur menggunakan model VGG16, Resnet50, dan InceptionV3. Serta membandingkan metode klasifikasi CNN dengan metode kombinasi CNN dan SVM untuk klasifikasi 4 kelas DR. |

### 2.3. Diabetic Retinopathy

*Diabetic Retinopathy* adalah suatu *mikroangiopati progresif* yang ditandai oleh kerusakan dan sumbatan pembuluh-pembuluh halus, meliputi *arteriol prekapiler* retina, kapiler-kapiler dan vena (Vaughan, 2000). Berkaitan dengan prediksi dan pengobatan, maka DR dibagi menjadi dua yaitu NPDR dan PDR (Freeman, 1998).



Gambar 2. 1. Ilustrasi Perbedaan Retina Normal dan *Retinopathy*

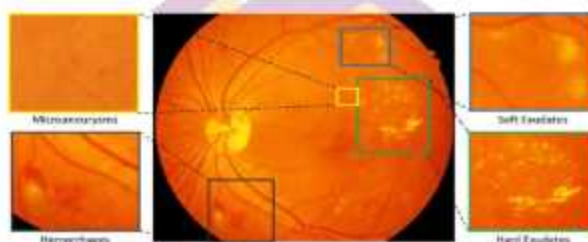
#### 2.3.1 *Non-Proliferative Diabetic Retinopathy (NPDR)*

NPDR adalah DR awal yang paling umum dijumpai. NPDR dibagi menjadi ringan, sedang dan berat. Tahap awal ditandai dengan adanya MA. MA merupakan pembengkakan pembuluh darah berukuran mikro yang pada pemeriksaan funduskopi akan terlihat seperti titik-titik kecil berwarna merah dekat pembuluh darah terutama *polus posterior* dengan diameter kurang dari 125 mikrometer (Meyerle, 2008).

Pada tahap berikutnya adalah dengan mendeteksi adanya HE (perdarahan). Sama sulitnya dengan mendeteksi MA, HE berukuran kecil dan tersebar pada daerah yang berbeda-beda serta memiliki bentuk titik, garis, dan bercak, terlihat seperti MA tetapi memiliki bentuk yang lebih besar. Para peneliti melakukan deteksi HE dengan mencari letak lesi merah dimana *hemorrhages* merupakan

komponen yang memiliki warna merah pada citra fundus retina (Biyani & Patre, 2018).

Tahap selanjutnya adalah munculnya EX yang terlihat seperti bercak berwarna kuning kaya lemak berbentuk bundar disekitar MA dan paling sering berpusat dibagian temporal makula. Dan SE yang tampak seperti bercak halus (*cotton wool spot*). Biasanya terletak dibagian tepi daerah *nonirigasi* dan dihubungkan dengan *iskemia* retina (L. Zhang et al., 2012). Perbandingan retina NPDR ditunjukkan pada gambar berikut:



Gambar 2. 2. Perbandingan retina NPDR

Gambar 2.2 adalah foto fundus retina yang menjelaskan ciri-ciri adanya NPDR. Bagian berwarna mengandung lesi retina berbeda yang berhubungan dengan DR. Bagian yang diperbesar menggambarkan keberadaan MA, SE, HE dan EX (Porwal et al., 2018).

### 2.3.2 Proliferative Diabetic Retinopathy (PDR)

Pada keadaan ini terdapat penyumbatan kapiler *microvascular* dan kebocoran plasma yang berlanjut, disertai iskemik pada dinding retina (*cotton wool spot*, *infarction* pada lapisan serabut saraf. Ciri khas dari stadium ini adalah *cotton wool spot*, *blot haemorrhage*, *Intraretinal Microvascular Abnormal (IRMA)*, munculnya pembuluh darah baru (*Neovaskularisasi*) pada retina yang biasanya

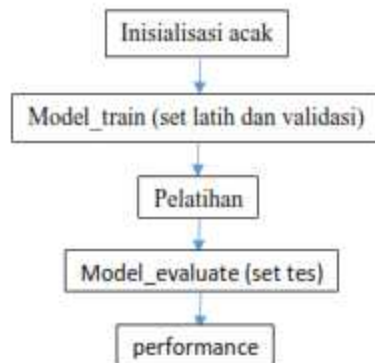
terletak dipermukaan jaringan. Tampak sebagai pembuluh yang berkelok-kelok, dalam, berkelompok, dan *ireguler*. Serta adanya edema retina dengan tanda hilangnya gambaran retina terutama daerah makula sehingga sangat mengganggu tajam penglihatan (Davis, 2006).

## 2.4. Preprocessing

Terkadang data tidak selalu dalam kondisi ideal untuk langsung diolah. Oleh karena itu diperlukan tahapan pengolahan ini untuk mentransformasikan data masukan menjadi data mengikuti tujuan yang akan dicapai. Pemilihan proses yang tepat di tahap *preprocessing* dapat meningkatkan kinerja klasifikasi. Tujuan utama dari *preprocessing* dan peningkatan citra adalah untuk mendapatkan gambar yang informatif secara visual, memudahkan pemrosesan selanjutnya dan langkah-langkah evaluasi gambar otomatis. Gambar medis merupakan salah satu bentuk gambar yang sulit diinterpretasikan oleh karena itu, pemrosesan awal sangat penting.

### 2.4.1 Pengelompokkan Data

Kumpulan data untuk *supervised learning* terdiri dari contoh-contoh, bergantung pada tugasnya. Setiap contoh dipasangkan dengan label membentuk sebuah kategori misalnya "NPDR" atau "PDR". Untuk melatih dan mengevaluasi model pembelajaran mesin, maka penting untuk membagi data menjadi tiga subset: set pelatihan, set validasi, dan set pengujian. Rasio yang umum digunakan misalnya data latih : data validasi : data uji adalah 70:15:15, 80:10:10, 60:20:20, dsb. Gambar berikut adalah langkah-langkah menggunakan tiga rasio data.



Gambar 2. 3. Penggunaan Tiga Set Rasio data Split

Data latih digunakan untuk melatih model. Data validasi digunakan untuk proses validasi model dan mencegah overfitting. Overfitting adalah kondisi dimana data yang digunakan saat pelatihan dapat belajar dengan sangat baik namun model tidak menganalisis pola trend dari keseluruhan data, dengan kata lain data bagus saat pelatihan namun buruk saat pengujian. Set validasi terdiri dari contoh yang belum pernah dilihat model pelatihan, jadi jika kita mendapatkan kinerja set validasi yang baik, artinya model kita telah belajar dengan baik. Data tes adalah data yang digunakan untuk menilai kinerja model menggunakan data yang belum pernah terlihat sebelumnya. Tidak ada patokan rasio yang tepat untuk jumlah masing-masing pengelompokan data. Karena hal itu adalah penilaian subjektif. Namun ada ketentuan umum sebagai berikut :

- Semakin banyak data latih artinya model kita melihat lebih banyak. Jumlah data latih tidak boleh lebih sedikit dari data validasi dan data tes karena akan menghasilkan performa pengujian yang buruk dengan kata lain tidak berfungsi.

- Semakin banyak data validasi baik karena dapat membantu membuat keputusan lebih baik tentang model mana yang terbaik.
- Lebih banyak data pengujian bagus karena memberi pemahaman yang lebih baik tentang seberapa baik model menggeneralisasi data yang tidak terlihat. Jika data pengujian tidak cukup maka penilaian akhir terhadap kemampuan generalisasi model mungkin tidak akurat.

Data apapun yang di tambahkan pada satu kumpulan harus diambil pada kumpulan lainnya. Untuk data kecil dapat menggunakan K-Fold Cross Validation.

#### 2.4.2 Data Augmentasi

Augmentasi data adalah teknik peningkatan data variasi, termasuk serangkaian manipulasi gambar operasi seperti pergeseran acak, membalik, rotasi acak, zoom acak, dan banyak lagi (He et al., 2016). Teknik tersebut bertujuan untuk memperluas kumpulan data secara *artificial* untuk membuat data baru (Wang & Perez, 2017). Seleksi teknik augmentasi data sangat berpengaruh pada hasil. Melatih model *deep learning neural network* pada data yang lebih banyak dapat menghasilkan model yang lebih cerdas untuk menggeneralisasi apa yang telah dipelajari ke data baru. Beberapa data augmentasi adalah :

- a. *Horizontal Flip* berarti membalik baris atau kolom pixel secara horizontal
- b. *Random brightness* adalah dengan mengubah kecerahan gambar, dengan lebih terang atau lebih gelap, dengan tujuan agar model dapat menggeneralisasi seluruh gambar yang dilatih pada tingkat kecerahan yang berbeda. Dengan parameter 0.0 sampai 0.2, dimana nilai 1.0 tidak merubah kecerahan, nilai kurang dari 0.1 mencerahkan, dan nilai lebih dari 1.0 adalah menggelapkan.



- c. *Rescale* digunakan untuk mengubah skala nilai piksel dari kisaran 0-255 ke kisaran 0-1 sebagai normalisasi sebelum pemrosesan lain menggunakan konstanta yang telah ditentukan sebelumnya.
- d. *Channel Shift Range* adalah pergeseran nilai acak berdasarkan nilai tetap dari saluran RGB dari gambar input

## 2.5. *Machine Learning*

Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel ditahun 1959. Menurut Arthur Samuel, *machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrogram yang jelas. *Machine learning* adalah cabang aplikasi dari *Artificial Intelligence* (Kecerdasan Buatan) yang fokus pada pengembangan sebuah sistem yang mampu belajar "sendiri" tanpa harus berulang kali di program oleh manusia. Aplikasi *Machine learning* membutuhkan Data sebagai bahan belajar (*training*) sebelum mengeluarkan *output*. Aplikasi sejenis ini juga biasanya berada dalam domain spesifik alias tidak bisa diterapkan secara umum untuk semua permasalahan (Samuel, 2000). Dilihat dari bagaimana *machine learning* belajar, maka dapat dibedakan menjadi *supervised learning* dan *unsupervised learning*.

### 2.5.1 *Supervised Learning*

Pembelajaran dengan pengawasan adalah algoritma pembelajaran mesin yang mempelajari suatu fungsi berdasarkan contoh yang sudah di ajarkan padanya (Russell, 2010). Algoritma ini seolah-olah sudah dilatih terlebih dahulu agar dapat melakukan prediksi maupun klasifikasi dengan menggunakan data latih

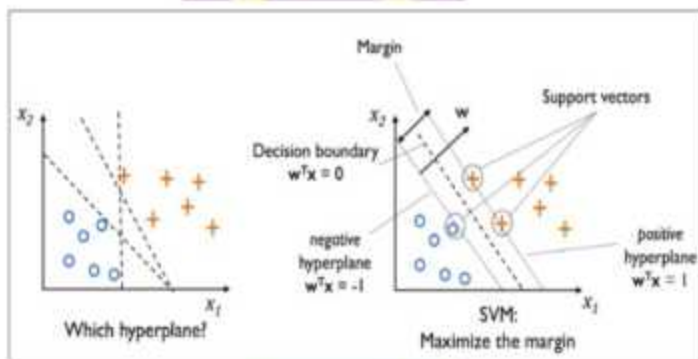
### 2.5.2 *Unsupervised Learning*

Pembelajaran tanpa pengawasan adalah algoritma pembelajaran mesin yang mencari pola yang sebelumnya tidak terdeteksi dalam kumpulan data sebelumnya dan dengan minimal pengawasan manusia (Hinton, 1999). Algoritma ini tidak menggunakan data latih atau data training untuk melakukan prediksi maupun klasifikasi

### 2.5.3 *Support Vector Machine (SVM)*

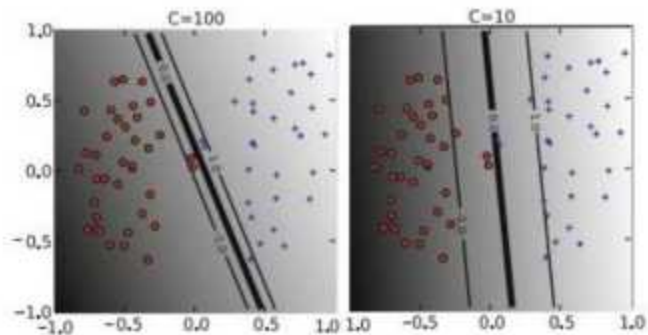
*Support Vector Machine (SVM)* pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition* (Vapnik, 1995). SVM merupakan salah satu metode dalam *supervised learning* yang biasanya digunakan untuk klasifikasi seperti pada *Support Vector Classification* dan *Support Vector Regression*. Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya (Cardoso et al., 2005). SVM digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas. *Hyperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai *line whereas*, fungsi yang digunakan untuk klasifikasi antar kelas dalam 3-D disebut *plane similarly*, sedangkan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut *hyperplane* (Cristianini & Shawe-Taylor, 2000).

*Hyperplane* yang diilustrasikan seperti Gambar 2.4, posisinya berada ditengah-tengah antara dua kelas, artinya jarak antara *hyperplane* dengan objek-objek data berbeda dengan kelas yang berdekatan (terluar) yang diberi tanda bulat kosong dan *positif*. Dalam SVM objek data terluar yang paling dekat dengan *hyperplane* disebut *support vector*. Objek yang disebut *support vector* paling sulit diklasifikasikan dikarenakan posisi yang hampir tumpang tindih (*overlap*) dengan kelas lain. Mengingat sifatnya yang kritis, hanya *support vector* inilah yang diperhitungkan untuk menemukan *hyperplane* yang paling optimal oleh SVM.



Gambar 2. 4. Hyperplane yang memisahkan dua kelas positif (+1) dan negatif (-1)

Pada ruang berdimensi tinggi inilah akan dicari *hyperplane* yang dapat memaksimalkan jarak (*margin*) antara kelas data.



Gambar 2. 5. Ilustrasi Pengaruh Parameter  $C$  (Ben-Hur & Weston, 2010)

Gambar 2. 5 menunjukkan pengaruh parameter  $C$  terhadap bentuk *hyperplane* dan hasil klasifikasi yang dilakukan pada dua kelas, yaitu kelas 1 (biru) dan kelas 2 (merah). Semakin besar nilai parameter  $C$ , maka semakin besar pula penalti yang diberikan kepada kesalahan klasifikasi. Nilai parameter  $C$  yang semakin kecil akan mengabaikan pengamatan yang dekat dengan *hyperplane* dan memperbesar margin. Pada saat nilai  $C$  besar, penalti yang besar diberikan kepada margin error. Seperti pada Gambar 2.5, saat  $C=100$ , dua pengamatan dari Kelas 1 yang berada paling dekat dengan *hyperplane* adalah *support vectors* dan mempengaruhi orientasi *hyperplane*. Saat nilai  $C$  semakin kecil ( $C=10$ ), dua pengamatan dari kelas 1 yang disebutkan sebelumnya berubah menjadi margin error, orientasi *hyperplane* berubah, dan memberikan margin yang lebih besar.

## 2.6. Deep learning

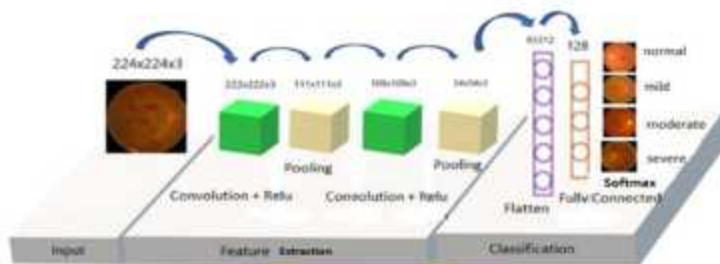
*Deep Learning* atau sering dikenal dengan istilah Pembelajaran Struktural Mendalam (*Deep Structured Learning*) atau Pembelajaran Hierarki (*Hierarchical learning*) adalah salah satu cabang dari ilmu pembelajaran mesin (*Machine Learning*) yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data

menggunakan sekumpulan fungsi *transformasi nonlinear* yang ditata berlapis-lapis dan mendalam (Schmidhuber, 2015). Algoritma pada *Deep Learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Dan algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya (Lecun et al., 2015).

## **2.7. Convolutional Neural Network (CNN)**

CNN merupakan salah satu algoritma dari *deep learning* yang merupakan hasil pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk mengolah data data yang besar dan diaplikasikan untuk mengolah data dua dimensi misalnya gambar atau suara (Maggiori et al., 2017). Penelitian awal yang menjadi dasar penemuan ini yaitu penelitian tentang *visual cortex* pada indera penglihatan kucing. *Visual cortex* pada hewan sangat *powerful* kemampuannya dalam sistem pemrosesan visual yang pernah ada (Hubel, 1970). Konsep tersebut kemudian dimatangkan oleh Yann LeCun dengan nama *LeNet* yang berhasil diterapkan pada penelitiannya mengenai pengenalan angka dan tulisan tangan (LeCun, 1990).

Arsitektur CNN secara umum terdiri dari *feature extraction* (*convolutional layer* dan *pooling layer*) dan *fully connected layer*. Gambaran umum arsitektur CNN berdasarkan *paper* asli dari LeCun dan Bengio ditunjukkan pada gambar 2.6.



Gambar 2. 6. Cara kerja CNN

CNN bekerja secara hierarki, sehingga *output* pada lapisan konvolusi pertama digunakan sebagai input pada lapisan konvolusi selanjutnya (Castelluccio et al., 2015). Pada proses klasifikasi terdiri dari *fully connected* dan fungsi aktivasi (*softmax*) yang *output*-nya berupa hasil klasifikasi (Katole et al., 2015). Berikut adalah penjelasan setiap tahapan CNN:

### 2.7.1 Convolution Layer

*Convolution layer* merupakan bagian tahap awal setelah input layer pada arsitektur CNN. Layer ini digunakan sebagai filter untuk mengekstraksi objek dari citra input. Filter ini berisi bobot yang digunakan untuk mendeteksi karakter dari objek seperti tepi, kurva, atau warna (Arrofiqoh & Harintaka, 2018). Filter diaplikasikan secara berulang sehingga menghasilkan serangkaian bidang *receptive*. Hasil dari bidang *receptive* berupa data tunggal.

### 2.7.2 RELU

ReLU merupakan operasi untuk mengenalkan *non-linearitas* dan meningkatkan representasi dari model. Layer ini dapat diibaratkan seperti *thresholding* atau sama halnya seperti fungsi aktivasi pada jaringan syaraf tiruan. Fungsi aktivasi ReLU adalah (Heaton, 2015)  $f(x) = \max(x, 0)$ . Dimana  $x$  adalah nilai pada *feature map*. Nilai *output* dari neuron bisa dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input adalah positif, maka *output* dari neuron adalah nilai input aktivasi itu sendiri (Kim et al., 2016).

### 2.7.3 Pooling Layer

*Pooling* atau *subsampling* adalah pengurangan ukuran matriks untuk mempercepat komputasi dan mengontrol terjadinya *overfitting*. Penggunaan *pooling layer* pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah *convolution layer* dengan *stride* yang sama dengan *pooling layer* yang bersangkutan (Springenberg, 2015). Terdapat dua macam *pooling* yang sering digunakan yaitu *average pooling* dan *max pooling* (Bejiga et al., 2017). Nilai yang diambil pada *average pooling* adalah nilai rata-rata sedangkan pada *max pooling* adalah nilai maksimal (Zhi et al., 2016).

### 2.7.4 Global Average Pooling

*Global Average Pooling* adalah operasi penggabungan yang dirancang untuk menggantikan lapisan *fully connected*. Tujuannya adalah untuk menghasilkan satu peta fitur untuk setiap kategori yang sesuai dari tugas klasifikasi di lapisan *convolusi* terakhir.

### 2.7.5 Fully Layer

*Fully-Connected Layer* merupakan kumpulan dari proses konvolusi. Lapisan ini mendapatkan input dari proses sebelumnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari lapisan ini adalah untuk menyatukan semua *node* menjadi satu dimensi (Albelwi & Mahmood, 2017). Dengan kata lain peran *Fully connection layer* adalah untuk mengekstraksi fitur lebih lanjut data keluaran CNN dan menghubungkan tahap ekstraksi fitur dengan klasifikasi softmax.

### 2.8. Model Pre-Trained

Model *pre-trained* adalah model yang sebelumnya telah dibuat dan dilatih pada *dataset* tertentu yang kemudian model tersebut dapat di implemetasikan pada dataset baru. Salah satu contoh model *pre-trained* adalah VGG16 (Simonyan & Zisserman, 2015), Inception V3 (Szegedy et al., 2016) dan Resnet50 (He et al., 2016).

### 2.9. Transfer learning

*Transfer learning* adalah memanfaatkan model pembelajaran yang telah dilatih sebelumnya pada data yang sangat besar (Siddiqi, 2019). Sehingga kita tidak perlu lagi melatih jaringan dari nol. Kita dapat mengambil pengetahuan dari data yang terlatih dan mengimplementasikannya pada data yang kita miliki dengan hanya mengubah pada tahap klasifikasi menyesuaikan jumlah kelas klasifikasi yang kita miliki. Untuk itu kita bisa membekukan semua konvolusi dan *fully* dengan penyatuan maksimal sehingga tidak mengubah bobotnya, hanya menyisahkan lapisan penghubung yang bebas yang dapat kita sesuaikan dengan pengklasifikaian



akhir, yang kemudian kita latih kembali, agar dapat berfungsi dengan lebih baik pada kumpulan data yang kita miliki itulah yang dimaksud dengan pembelajaran transfer.

### 2.9.1 Inception V3

Christian Szegedy mengembangkan jaringan yang disebut inceptionnet (Szegedy et al., 2016). Konsep dari arsitektu Inception V3 dari GoogLeNet dirancang untuk dapat bekerja dengan baik keterbatasan memori dan anggaran computer (Szegedy et al., 2015). Untuk mencapai tujuan tersebut penulis GoogLeNet menggunakan penggabungan saluran dari konvolusi  $1 \times 1$ , konvolusi  $3 \times 3$ , konvolusi  $5 \times 5$ , dan penggabungan bersama dengan menjaga tinggi dan lebar setiap saluran tidak berubah. Untuk menghindari peningkatan eksponensial dari operasi matematika yang harus dilakukan untuk penggabungan saluran, lapisan "bottleneck"  $1 \times 1$  yang memiliki kedalaman saluran lebih dangkal diterapkan sebelum menerapkan konvolusi  $3 \times 3$  atau  $5 \times 5$ . Pengenalan lapisan "bottleneck" ini mengurangi jumlah matematika operasi untuk konvolusi tertentu dengan faktor 10. Sebagai hasil dari penggabungan saluran dan lapisan "bottleneck" konvolusi  $1 \times 1$ , GoogleNet hanya menggunakan 5 juta parameter, 12x lebih kecil dari pendahulunya AlexNet (Alex Krizhevsky, Ilya Sutskever, 2012), yang menggunakan 60 juta parameter. Lebih lanjut, VGGNet menggunakan sekitar 3x lebih banyak parameter daripada AlexNet. Dibandingkan dengan GoogleNet (Szegedy et al., 2015), InceptionNet V3 menggunakan konvolusi pemfaktoran strategi untuk lebih meningkatkan efisiensi komputasi. Gambar 4 menunjukkan

modul awal InceptionNet V3 memanfaatkan faktorisasi spasial ke dalam konvolusi asimetris untuk menghemat biaya komputasi lebih lanjut.

### 2.9.2 VGG16

VGG16 telah dilatih untuk melakukan klasifikasi pada 1000 kelas dengan dataset Imagenet yang terdiri dari 1.200.000 gambar untuk data train dan 100.000 gambar untuk data test. Jaringan ini menggunakan *Convolutional* dengan *kernel* 3x3 yang ditumpuk satu sama lain untuk meningkatkan kedalaman. Sedangkan *Max-Pooling* digunakan untuk mengurangi volume. Kelebihan VGG16 adalah memiliki fitur yang menarik dan keserhanaan arsitektur (Simonyan & Zisserman, 2015).

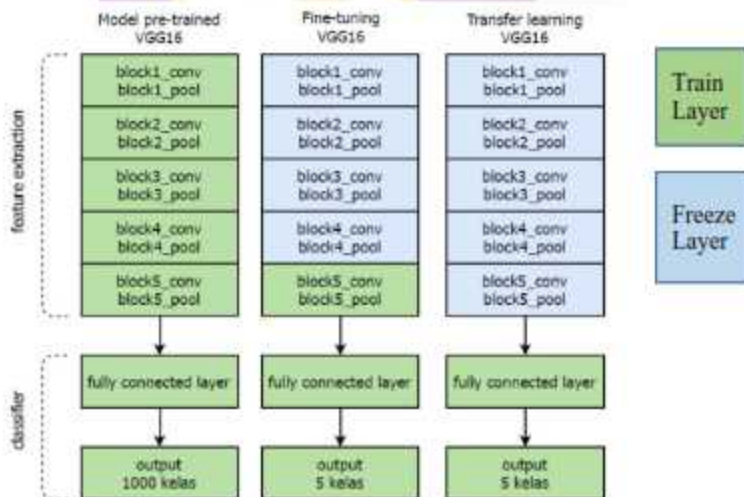
### 2.9.3 Resnet50

Residual Neural Network (ResNet) yang dikemukakan oleh (He et al., 2016) telah berhasil melatih 152 jaringan saraf mendalam dan berhasil memenangkan kejuaraan ILSVRC 2015. Pada penelitiannya dia berhasil mendapatkan klasifikasi tingkat kesalahan 3,57% untuk 5 kelas teratas walaupun jumlah parameternya kurang dari model VGG. Inti dari model Resnet ini adalah *High Way Nets* memberiarakan beberapa masukan melewati lapisan tanpa pandang bulu untuk mengintegrasikan aliran informasi untuk menghindari hilangnya informasi yang di transfer dalam lapisan dan masalah hilangnya gradien. Selain itu juga menekan beberapa noise dengan membuat rata-rata model dengan menyeimbangkan akurasi pelatihan dan generalisasi. Dengan cara menambahkan lebih banyak data label, untuk mencapai akurasi pelatihan yang lebih tinggi. Gagasan Resnet berasal dari munculnya degradasi kedalaman CNN. Dimana akurasi meningkat pada awalnya

dan kemudian mencapai Batasan, sementara jika semakin meningkatkan kedalaman akan menurunkan akurasi.

## 2.10. Fine-tuning

*Fine-tuning* adalah Teknik yang digunakan untuk menggunakan Kembali model pre-trained dengan cara membekukan beberapa lapisan (*freeze layer*) serta bersama-sama melatih model *classifier* baru dan *layer* yang tidak beku (*unfreeze layer*). Seperti contoh pada Fine-Tuning VGG16 untuk melakukan klasifikasi level NPDR menjadi *normal*, *mild*, *moderate*, dan *severe*. VGG16 terdiri dari 5 block *convolutional layer* sebagai *feature extraction*. Perbedaan transfer learning, fine tuning, dan model pretrained ditunjukkan pada gambar 2. 7 di bawah ini.



Gambar 2. 7. Pre-Trained VGG16 dan Fine Tuning

## 2.11. Multi Class Confusion Matrix

*Confusion matrix* adalah tabel yang digunakan untuk mengukur kinerja hasil klasifikasi dari model prediksi. Tabel ini menjelaskan performa model klasifikasi

pada rangkaian data pengujian benar atau salah. Nilai *True Negative* (TN) adalah data yang di klasifikasi dengan tepat sebagai keluaran negatif atau salah. *True Positive* (TP) adalah data yang diklasifikasi dengan tepat sebagai keluaran positif atau benar. *False Positive* (FP) adalah data yang diklasifikasi dengan kurang tepat apabila keluaran berupa positif atau benar. *False Negative* (FN) adalah data yang diklasifikasi dengan kurang tepat. Untuk perhitungan *confusion matrix* multikelas akan mengacu pada tabel 2.2, untuk masing-masing kelas  $i$  digunakan rumus skor seperti pada persamaan 1 untuk mencari nilai True Positif (TP), True Negatif (TN), False Positif (FP), dan False Negatif (FN) (Manliguez, 2016).

$$\begin{aligned}
 TP_{all} &= \sum_{j=1}^n X_{jj} \\
 TN_i &= \sum_{j=1}^n \sum_{k \neq i} X_{jk} \\
 FP_i &= \sum_{j=1}^n X_{ji} \\
 FN_i &= \sum_{j \neq i} X_{ij}
 \end{aligned} \tag{1}$$

Tabel 2. 2. *Multi Class Confusion Matrix*

|           |         | Predicted Data  |                 |                 |     |                 |
|-----------|---------|-----------------|-----------------|-----------------|-----|-----------------|
|           |         | Class 1         | Class 2         | Class 3         | ... | Class n         |
| True Data | Class 1 | X <sub>11</sub> | X <sub>12</sub> | X <sub>13</sub> | ... | X <sub>1n</sub> |
|           | Class 2 | X <sub>01</sub> | X <sub>11</sub> | X <sub>21</sub> | ... | X <sub>2n</sub> |
|           | Class 2 | X <sub>02</sub> | X <sub>12</sub> | X <sub>22</sub> | ... | X <sub>3n</sub> |
|           | ...     | ...             | ...             | ...             | ... | ...             |
|           | Class n | X <sub>n1</sub> | X <sub>n2</sub> | X <sub>n3</sub> | ... | X <sub>nn</sub> |

*Confusion matrix* ini adalah dasar untuk menghitung nilai akurasi sensitivitas, dan Spesifity untuk setiap kelas  $i$  dijelaskan pada persamaan 2,3, 4.

Rumus untuk mendapatkan nilai akurasi bisa dilihat pada persamaan 2 di bawah ini :

$$\text{Akurasi} = \frac{TP_{all}}{\text{Total Number of Testing Entris}} \quad (2)$$

Rumus untuk mencari sensitivitas (SS) dapat dilihat pada persamaan 3 di bawah ini:

$$SS_i = \frac{TP_{all}}{TP_{aai} + FN_i} * 100\% \quad (3)$$

Persamaan (3) merupakan perhitungan nilai sensitivitas pada setiap kelas dan rata-rata pada semua kelas. Evaluasi sensitivitas digunakan apabila lebih memilih nilai FP daripada FN.

Rumus untuk menghitung spesifitas (SP) tiap kelas dan seluruh kelas dapat dilihat pada persamaan 4 di bawah ini :

$$SP_i = \frac{TN_{all}}{TN_{aai} + FPI} * 100\% \quad (4)$$

Persamaan (4) merupakan kebenaran memprediksi negatif dibandingkan dengan keseluruhan data negatif.

## BAB III METODE PENELITIAN

### 3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini merupakan penelitian experimental. Eksperimen yang dilakukan adalah dengan mengkombinasikan metode CNN dan metode SVM untuk mendeteksi dan klasifikasi NPDR.

Dilihat dari segi sifatnya penelitian ini adalah penelitian bersifat deskriptif dimana penelitian ini nantinya akan menjelaskan dampak penggunaan metode SVM terhadap metode CNN dalam mendeteksi dan klasifikasi NPDR.

Pada penelitian ini menggunakan pendekatan kuantitatif, dimana penelitian ini akan menghitung akurasi, sensitivitas dan spesifitas metode kombinasi CNN dan SVM dalam mendeteksi dan klasifikasi DNPR.

### 3.2. Metode Pengumpulan Data

Dalam penelitian ini, metode pengumpulan data dilakukan terhadap objek citra fundus retina dari data publik. *Dataset* yang digunakan dalam penelitian ini adalah *messidor database* berupa citra fundus retina yang mengandung tiga komponen warna yaitu merah (R), hijau (G) dan biru (B). *Database* ini dirancang untuk keperluan penelitian ilmiah dalam deteksi dan klasifikasi DR. *Database messidor* adalah *database* yang disediakan untuk umum sejak 2014. Basis data *messidor* dihasilkan oleh OPHDIAT © jaringan telemedical dari penyaringan DR.

### 3.3. Metode Analisis Data

Analisis data dilakukan setelah melakukan pengujian metode dengan menghitung akurasi, sensitivitas dan spesifitas metode CNN dan metode kombinasi CNN dan SVM dalam mendeteksi dan klasifikasi DNPR. Hasil dari pengujian ini adalah klasifikasi NPDR berdasarkan level keparahannya yaitu NPDR ringan, sedang dan parah.

### 3.4. Alur Penelitian

Alur penelitian pada penelitian ini adalah sebagai berikut, dan dijelaskan melalui bagan alur penelitian pada gambar 3.1.

#### 1. Pengumpulan Data

Data pada penelitian ini adalah data sekunder yang merupakan open data yang dapat diakses oleh umum dan dapat di unduh pada website <http://www.adcis.net/en/third-party/messidor/>. Database messidor diperoleh dari tiga departemen oftalmologi dan telah dikelompokan / label oleh ahlinya berdasarkan level keparahan DR. Gambar dalam format TIFF dengan dimensi 1440 \* 960, 2240 \* 1488 atau 2304 \* 1536 piksel (Decenci re et al., 2014).

#### 2. Menambah Volume Data

CNN bekerja dengan baik pada dataset yang besar, namun jika dataset yang tersedia kecil, maka kita bisa menambah volume data dengan cara mentransformasi data yang sudah ada yang dikenal dengan teknik augmentasi data. Teknik augmentasi menambah volume data yang digunakan adalah *horizontal flip*. Peningkatan data dengan augmentasi yang sebelumnya

dilakukan pada penelitian (Agustin et al., 2020) untuk klasifikasi retina normal dan DR.

### 3. Preprocessing

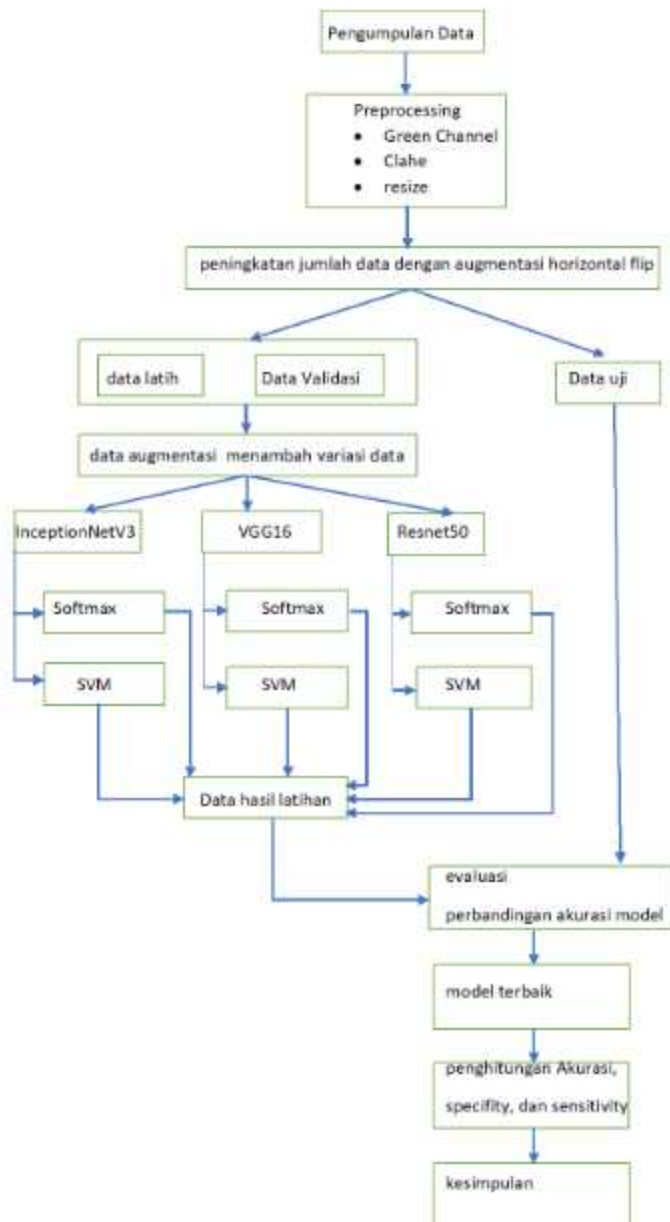
Tahap *preprocessing* diperlukan untuk memperbaiki citra, beberapa tahap dalam proses *preprocessing* yang akan dilakukan antara lain:

- a. Ekstraksi *green chanel* : dipilih karena dapat menampilkan kontras maksimum antara latar belakang dan lesi. Sedangkan saluran merah dan biru cenderung mengandung lebih banyak kebisingan dibandingkan dengan saluran hijau. Selain itu dalam chanel hijau ini pembuluh darah dapat terlihat lebih jelas (T. A. Soomro, T. Mahmood Khan,2018)
- b. CLAHE : hasil citra *green chanel* akan diolah kembali dengan menggunakan CLAHE untuk meningkatkan kontras. CLAHE dapat memperjelas penampang pembuluh darah lebih baik dari teknik peningkatan kontras lainnya (Erwin,2020)
- c. Selanjutnya proses *resize* yang bertujuan untuk menyamakan ukuran piksel dari citra yang akan diolah agar cocok untuk menjadi input persegi bagi proses ekstraksi fitur dalam arsitektur CNN selanjutnya.

### 4. Train data split

Selanjutnya dataset dibagi menjadi data latih, data validasi, dan data uji dengan pembagian data secara acak menggunakan *train\_test\_split* yang disediakan oleh *Sklearn* dari *library Python*.





Gambar 3. 1. Alur Penelitian

#### 5. Variasi data latih dengan augmentasi

Pada tahap ini adalah menggunakan data augmentasi hanya sebagai variasi data latih tanpa penambahan jumlah data. Data augmentasi yang digunakan adalah *random brightness*, *random zoom*, *channel shift* yang akan dijelaskan pada bab sebelumnya.

#### 6. Skenario penelitian

Karena jenis penelitian ini bersifat eksperimen maka alur penelitian disediakan langkah skenario. Dengan menggunakan metode CNN dan menggunakan metode kombinasi CNN dan SVM penulis ingin mencari akurasi, spesifisitas dan sensitivitas yang tertinggi. Layer tiga pada CNN atau yang biasa disebut sebagai *fully layer* adalah layer yang digunakan untuk klasifikasi. Dalam penggunaan klasifikasi metode kombinasi CNN dan SVM, maka *fully layer* ini tidak digunakan dan akan digantikan dengan klasifikasi pada metode SVM.

#### 7. Data hasil latih

Setelah menentukan arsitektur mana yang terbaik dari skenario yang ada kita mulai melakukan percobaan menggunakan dataset citra yang telah di *preprocessing* sebelumnya. Data hasil latih ini merupakan hasil skenario terbaik untuk menentukan akurasi, sensitivitas dan spesifitas tertinggi. Model klasifikasi tersebut menjadi sebuah model yang direkomendasikan pada tahap evaluasi atau pengujian data

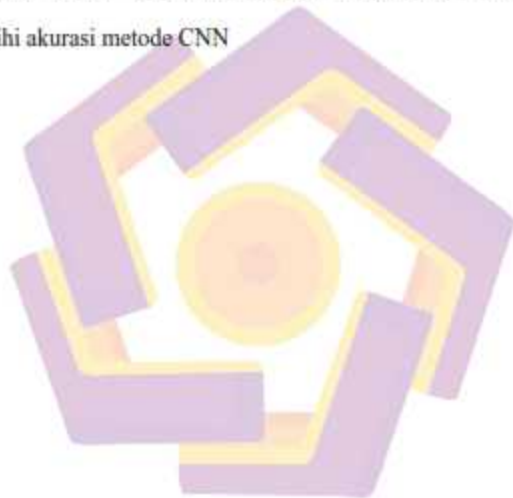
#### 8. Evaluasi

Melakukan perbandingan akurasi pada semua skenario yang telah dilakukan.

Perhitungan nilai akurasi, sensitivitas dan spesifitas dengan menggunakan *Confusion Matrix*.

#### 9. Kesimpulan

Tahap ini menyajikan hasil dari penelitian. Hasil penelitian berupa data fakta yang dihasilkan dari nilai *confusion matrix* terkait dengan metode klasifikasi CNN dan metode klasifikasi kombinasi CNN dan SVM. Apakah metode kombinasi efektif digunakan dengan mendapatkan akurasi yang tinggi melebihi akurasi metode CNN



## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Tahap Membangun Dataset

##### 4.1.1 Pengumpulan Data

Data yang digunakan berupa citra fundus retina yang terdiri dari empat kelas level keparahan DR, yaitu normal, *mild*, *moderate*, dan *severe*. Data gambar yang diperoleh merupakan data sekunder yang disediakan oleh messidor database (Decenciére et al., 2014). Messidor adalah singkatan dari *Methods to Evaluate Segmentation and Indexing Techniques in the field of Retinal Ophthalmology* yang merupakan program penelitian yang di danai oleh Kementerian Riset dan Pertahanan Prancis dalam program TECHNO-VISION 2004. Dataset ini dapat di unduh gratis hanya untuk kepentingan penelitian. Data gambar dalam penelitian ini berjumlah 984 dengan rincian seperti pada tabel 4. 1.

Tabel 4. 1. Pembagian Data

| Diabetic Severity | total |
|-------------------|-------|
| Normal            | 381   |
| Mild              | 153   |
| Moderate          | 196   |
| Severe            | 254   |

##### 4.1.2 Pengelompokan Data

Pengelompokan data dilakukan dengan menggunakan rasio perbandingan 70:15:15 untuk masing-masing data latih : data validasi : data uji. Pembagian data dilakukan secara acak menggunakan *train\_test\_split* yang disediakan oleh *Sklearn* dari *library Python* yang menawarkan berbagai fitur untuk pemrosesan data. Alasan digunakannya rasio perbandingan data ini telah dijelaskan sebelumnya di landasan

teori pada bab II. *Script* yang digunakan untuk melakukan *train\_test\_split* pada citra fundus retina adalah sebagai berikut.

```

1. from sklearn.model_selection import train_test_split
2. from keras.utils import to_categorical
3. train_ratio = 0.70
4. validation_ratio = 0.10
5. test_ratio = 0.10
6. X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1 - train_ratio, random_state=1)
7. X_val, X_test, y_val, y_test = train_test_split(X_test, y_test, test_size=test_ratio/(test_ratio + validation_ratio), random_state=1)

```

Baris 1-2, import module yang digunakan untuk *train\_test\_split*.

Baris 3-4, menentukan porsi rasio perbandingan data.

Baris 6-7, membagi gambar menjadi 3 bagian yang acak sesuai dengan rasio yang telah ditentukan.

#### 4.1.3 Perlakuan Terhadap Data

Kami mengembangkan 984 gambar menjadi 3.936 gambar menggunakan *horizontal Flip* untuk mengurangi kesalahan generalisasi seperti yang diterangkan oleh (Goodfellow et al., 2016). Selain itu data augmentasi juga digunakan untuk meningkatkan variasi data pada data latih dengan tujuan untuk menekan *overfitting*. Dengan menggunakan *Random Brightness*, *Channel Shift* dan *rescale*

#### 4.2. Tahap Proses Analisis

Untuk menganalisis metode kombinasi CNN dan SVM dalam klasifikasi NPDR, kami akan melakukan beberapa skenario percobaan untuk mendapatkan nilai akurasi dan sensitivitas yang tinggi. Namun kami juga akan menguji akurasi dari metode klasifikasi CNN untuk mendapatkan perbandingan hasil, apakah metode klasifikasi kombinasi CNN dan SVM memang jauh lebih baik dari pada

metode klasifikasi dari CNN sendiri, sehingga metode kombinasi patut untuk direkomendasikan. Skenario yang kami gunakan dalam penelitian adalah sebagai berikut:

- a. Skenario pertama adalah bagaimana membangun model ekstraksi fitur dengan melakukan pengujian pada:
  - *Transfer learning* VGG16, Resnet50, dan InceptionV3
  - *Fine Tuning* VGG16
  - *Pretrained model* VGG16, Resnet50, dan InceptionV3
- b. Skenario kedua adalah membandingkan klasifikasi menggunakan CNN *Softmax* pada arsitektur VGG16, Resnet50, dan InceptionV3
- c. Skenario ketiga adalah membandingkan klasifikasi menggunakan CNN SVM pada arsitektur VGG16, Resnet50, dan InceptionV3
- d. Skenario keempat adalah mengoptimalisasi model kombinasi CNN dan SVM

#### **4.3. Tahap Implementasi Praproses**

Pada subbab ini akan dijabarkan implementasi fungsi-fungsi pada tahap praproses, mulai dari pembacaan data hingga penyimpanan data akhir. Juga disertakan kode sumber dari masing-masing fungsi. Seluruh fungsi pada seluruh program menggunakan GPU yang disediakan langsung pada *google collabs* yaitu *coding environment* bahasa pemrograman *Python* yang merupakan salah satu produk Google berbasis *cloud* yang dapat diakses secara gratis. Selain itu pada *google colabs* ini juga telah menyediakan GPU dengan spek tinggi yaitu GPU Tesla, RAM 12GB, Disk 300GB yang dapat diakses secara gratis.

### 4.3.1 Implementasi Fungsi Load Data

Pertama-tama kita akan mengimport modul yang diperlukan seperti pada baris 1-3. Modul *os* menyediakan berbagai fungsi untuk berinteraksi dengan system seperti pada baris 5. Import *drive* digunakan untuk mengintegrasikan google colabs dengan drive kita untuk menjalankannya kita gunakan script pada baris 3. Modul *re* menyediakan alat ekspresi reguler untuk pemrosesan string lanjutan. Modul *glob* menyediakan fungsi untuk membuat daftar berkas dari pencarian *wildcard* di direktori, untuk menemukan lokasi direktori digunakan kode pada baris 6. Sebelumnya telah di definisikan letak folder yang kita gunakan pada baris 4.

```

1. import os, re, glob, cv2
10. from google.colab import drive
11. drive.mount('/content/drive')
12. PATH = '/content/drive/My Drive/Colab Notebooks/AI/4class/training/'
13. dirs = os.listdir(PATH)
14. for pic in glob.glob(PATH+'**/*'):

```

### 4.3.2 Implementasi *Preprocessing*

#### 4.3.2.1 *Green Channel*

RGB (*red, green, blue*) adalah warna dasar yang dapat dipahami oleh komputer. *Green channel* digunakan pada keseluruhan skenario dalam penelitian ini. Untuk hanya mengambil kernel hijau saja digunakan *script* pada baris 2-4. Baris 1 digunakan untuk memanggil *c2* dari *OpenCV*. Modul *OpenCV* adalah modul yang digunakan untuk mengolah gambar.

```

1. in = cv2.imread(pic)
2. green_image = in.copy()
3. green_image[:, :, 0] = 0
4. green_image[:, :, 2] = 0

```

#### 4.3.2.2 CLAHE (*Contrast Limited Adaptive Histogram Equalization*)

Clahe digunakan sebagai variable tetap bersama dengan *green channel* dalam penelitian ini. Untuk mengimplementasikan Clahe adalah dengan menambahkan *script* dibawah ini.

```

5. lab_in = cv2.cvtColor(green_image , cv2.COLOR_BGR2LAB)
6. l,a,b =cv2.split(lab_in)
7. clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
8. clahe_in = clahe.apply(l)
9. updated_lab_in2 = cv2.merge((clahe_in,a,b))
10. CLAHE_in = cv2.cvtColor(updated_lab_in2, cv2.COLOR_LAB2BGR)

```

#### 4.3.3 Implementasi Data Augmentasi

Untuk melakukan augmentasi penulis memanfaatkan pustaka deep learning keras yang menyediakan kemampuan menggunakan augmentasi data secara otomatis. Sehingga kita perlu memanggil fungsi tersebut seperti pada baris 1. Dalam penelitian ini data augmentasi yang digunakan dibagi menjadi dua tujuan, pertama untuk meningkatkan volume data dan kedua hanya untuk meningkatkan variasi data latih. Teknik augmentasi yang digunakan untuk meningkatkan volume data dapat dilihat pada kode di bawah ini. Data augmentasi yang digunakan adalah *horizontal flip* yang cocok dengan database yang penulis gunakan. Disini penulis membuat iterasi untuk dataset gambar. *Iterator* akan mengembalikan satu kumpulan gambar yang ditambah untuk setiap iterasi yang dimuat dalam memori melalui fungsi *flow ()* pada baris 8.

```

a. from keras.preprocessing.image import ImageDataGenerator
b. from keras.preprocessing.image import img_to_array
c. import numpy as np
d. from numpy import expand_dims
e. data = img_to_array(CLAHE_in)
f. samples = expand_dims(data, 0)
g. datagen = ImageDataGenerator(horizontal_flip=True)

```



```

h. it = datagen.flow(samples, batch_size=1)
i. for a in range(4):
j.     batch = it.next()
k.     img1 = batch[0].astype('uint8')

```

Baris 1-4, import module yang digunakan

Baris 5, mengkonversikan ke array numpy

Baris 6, memperluas dimensi menjadi satu sampel

Baris 7, membuat generator data augmentasi gambar

Baris 8-9, menyiapkan iterasi

Baris 10, menghasilkan kumpulan gambar

Baris 11, mengkonversikan ke bilangan integer

Sedangkan data augmentasi yang digunakan untuk menambah variasi data

latih ditunjukkan pada kode dibawah ini

```

2. datagen = ImageDataGenerator(
3.     brightness_range=[0.2, 1.2],
4.     samplewise_center=True,
5.     channel_shift_range=100,
6.     rescale=1.0/255.0
7. )
8. datagen.fit(X_train)

```

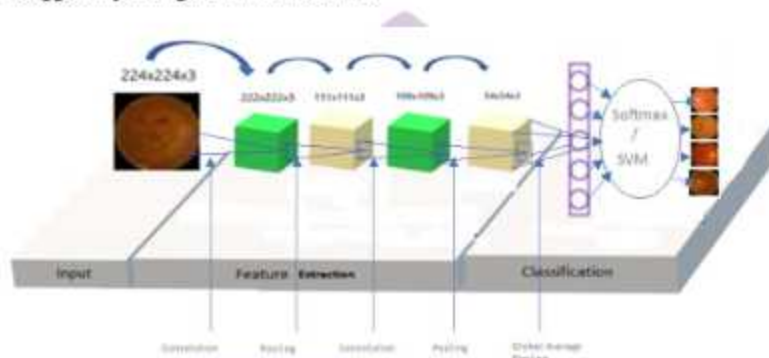
masih sama dengan sebelumnya, disini juga memanfaatkan keras data generator.

Baris 3-6 adalah jenis augmentasi yang digunakan. Setelah konfigurasi datagenerator dibuat, selanjutnya kita transformasikan dengan memanggil fungsi fit() untuk meneruskannya ke set data pelatihan seperti pada baris 7.

#### 4.4. Tahap membangun Model

Pembahasan pada subbab ini akan menjabarkan implementasi fungsi pada tahap pembangunan model CNN. Dalam penelitian ini, kita menggunakan model *transfer learning* VGG16, InceptionNet dan Resnet50. Dan menggunakan

klasifikasi dari CNN dan SVM. Secara umum model yang dibangun ditunjukkan pada gambar 4. 1. Input layer adalah data input dengan ukuran 224x224px. Fitur ekstraksi layer yang digunakan adalah model arsitektur transfer learning yang telah disebutkan sebelumnya. Pada bagian output layer adalah layer klasifikasi yaitu lapisan *fully*. Pada lapisan ini model klasifikasi dapat disesuaikan dengan keinginan. Pada penelitian ini kita akan menggunakan softmax dari CNN atau bisa menggantinya dengan klasifikasi SVM.



Gambar 4. 1. Arsitektur Model Klasifikasi yang Dibangun

Konfigurasi Semua skenario diuji dengan penambahan data augmentasi *horizontal flip*, *channel green* dari RGB, dan CLAHE dengan parameter seperti pada tabel 4. 2.

Tabel 4. 2. CNN *Train* Parameter

| Optimizer     | SGD                      |
|---------------|--------------------------|
| Learning Rate | 1e-3                     |
| Loss          | categorical crossentropy |
| Metric        | accuracy                 |
| Aktivasi      | softmax                  |
| Dropout       | 0.6                      |
| Epoch         | 70                       |
| Batch Size    | 82                       |

## 4.5. Tahap Implementasi Skenario Pertama

### 4.5.1 Implementasi Transfer Learning

Pada tahap ini adalah melakukan perbandingan hasil dari *transfer learning* model Resnet50, InceptionV3, dan VGG16. Modul yang digunakan pada setiap model hampir sama, yang berbeda hanyalah pada modul import model yang digunakan. Berikut adalah modul import yang digunakan oleh masing-masing model :

```
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.resnet50 import ResNet50
```

Kemudian memanggil fungsi model sesuai dengan arsitektur yang akan digunakan, dengan input baru yang kami berikan yaitu 224x224 px dan tanpa menyertakan model fully bawaan dari arsitektur transfer learning dengan parameter *False* pada *include\_top*. Misalkan untuk memanggil model arsitektur InceptionV3 dengan menggunakan modul berikut, dan begitu pula untuk model lainnya disesuaikan dengan Namanya.

```
base_model = InceptionV3(include_top=False, input_shape=(224, 224, 3))
```

Kode selanjutnya adalah membuat lapisan *fully* yang baru sesuai dengan kebutuhan. Parameter pada *layer.trainable* disetting dengan *False* artinya kita hanya membekukan semua lapisan kecuali lapisan *fully* yang baru kita buat.

```
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
dropOutCus = Dropout(0.6)(x)
predictions = Dense(4, activation='softmax')(dropOutCus)
model = Model(inputs=base_model.input, outputs=predictions)
for layer in base_model.layers:
    layer.trainable = False
```

```
model.summary()
```

Sebagai contoh berikut adalah parameter dari transfer learning model Resnet50. dengan tambahkan pada lapisan fully yang telah kami buat sebelumnya. Total parameter yang tidak dilatih berjumlah 23,587,712 dari total parameter 25,689,988.

| Layer (type)  | Output Shape       | Param # | Connected to           |
|---|--------------------|---------|------------------------|
| conv5_block3_out (Activation)                       | (None, 7, 7, 2048) | 0       | conv5_block3_add[0][0] |
| global_average_pooling2d_2 (GlobalAveragePooling2D) | (None, 2048)       | 0       | conv5_block3_out[0][0] |
| dense_4 (Dense)                                     | (None, 1024)       | 2058176 |                        |
| global_average_pooling2d_2[0][0]                    |                    |         |                        |
| dropout_2 (Dropout)                                 | (None, 1024)       | 0       | dense_4[0][0]          |
| dense_5 (Dense)                                     | (None, 4)          | 4100    | dropout_2[0][0]        |

Total params: 25,689,988  
 Trainable params: 2,102,276  
 Non-trainable params: 23,587,712

#### 4.5.2 Fine-Tuning VGG16

*Fine-tuning* berarti menggunakan kembali kemampuan klasifikasi VGG16 pada dataset yang berbeda (dataset DR). Untuk melakukan klasifikasi DR menggunakan *fine-tuning* maka beberapa *convolutional layer* ada yang dilatih kembali (*trainable layer*). kami mencoba untuk melakukan beberapa skenario untuk melakukan pelatihan ulang pada tiap blok, misalkan pada *convolutional layer block 3* sampai *5 (trainable layer)* dan membekukan (*freeze layer*) *convolutional block 1* dan *2*. Kami juga merancang *fully connected layer* sesuai dengan model pelatihan yang akan kami buat. Pada bagian output, kami membagi menjadi 4 kelas, sesuai dengan jumlah kelas yang kami butuhkan.

Rincian dari setiap *convolutional layer* pada model *fine tuning* VGG16 dengan input 224x224 px ditunjukkan pada Tabel 4. 3. Baris yang berwarna biru

merupakan *convolutional layer* yang kami bekukan (*freeze layer*), sedangkan baris berwarna hijau merupakan *convolutional layer* yang kami latih (*trainable layer*).

Tabel 4. 3. arsitektur mode fine tuning VGG16

| Layer Name   | Type         | VGG16         |         |
|--------------|--------------|---------------|---------|
|              |              | Output Shape  | Param   |
| Input 1      | InputLayer   | 224, 224, 3   | 0       |
| block1_conv1 | Conv2D       | 224, 224, 64  | 1792    |
| block1_conv2 | Conv2D       | 224, 224, 64  | 36928   |
| block1_pool  | MaxPooling2D | 112, 112, 64  | 0       |
| block2_conv1 | Conv2D       | 112, 112, 128 | 73856   |
| block2_conv2 | Conv2D       | 112, 112, 128 | 147584  |
| block2_pool  | MaxPooling2D | 56, 56, 128   | 0       |
| block3_conv1 | Conv2D       | 56, 56, 256   | 295168  |
| block3_conv2 | Conv2D       | 56, 56, 256   | 590080  |
| block3_conv3 | Conv2D       | 56, 56, 256   | 590080  |
| block3_pool  | MaxPooling2D | 28, 28, 256   | 0       |
| block4_conv1 | Conv2D       | 28, 28, 512   | 1180160 |
| block4_conv2 | Conv2D       | 28, 28, 512   | 2359808 |
| block4_conv3 | Conv2D       | 28, 28, 512   | 2359808 |
| block4_pool  | MaxPooling2D | 14, 14, 512   | 0       |
| block5_conv1 | Conv2D       | 14, 14, 512   | 2359808 |
| block5_conv2 | Conv2D       | 14, 14, 512   | 2359808 |
| block5_conv3 | Conv2D       | 14, 14, 512   | 2359808 |
| block5_pool  | MaxPooling2D | 7, 7, 512     | 0       |

Jumlah parameter yang dilatih dengan metode fine-tuning ditunjukkan pada

Tabel 4.4. Total parameter untuk *trainable parameter* dan *Non-trainable parameter* bergantung pada jumlah blok layer yang di latih, sesuai dengan skenario yang penulis buat. Berikut adalah salah satu contoh dari skenario kami ditunjukkan pada tabel 4. 4.

Tabel 4. 4. Parameter Fine Tuning VGG16

| Fine-Tuning VGG16       |            |
|-------------------------|------------|
| Total parameter         | 14,714,688 |
| Trainable parameter     | 14,454,528 |
| Non-trainable parameter | 260,160    |

Tahap selanjutnya adalah melakukan training model *fine tuning* VGG16 dengan meng-import modul-modul yang diperlukan untuk melakukan pelatihan dengan menggunakan CNN seperti yang ditunjukkan pada baris 1-8.

```
1. import numpy as np
2. import tensorflow.keras as tf
3. from keras.preprocessing.image import ImageDataGenerator
4. from keras.models import Sequential
5. from keras.layers import Dropout, Flatten, Dense
6. from keras.applications.vgg16 import VGG16
7. from keras.optimizers import SGD
8. import matplotlib.pyplot as plt
```

Model *fine tuning* VGG16 yang kami gunakan tanpa mengikutsertakan *fully connected layer* karena kami akan membuat sendiri *layer fully* tersebut sesuai dengan kebutuhan kami.

```
9. prior = keras.applications.VGG16()
10.     include_top=False,
11.     weights='imagenet',
12.     input_shape=(224, 224, 3)
13. )
```

Parameter *trainable* pada *layer* digunakan untuk membekukan bobot lapisan tertentu yang dapat kita atur menjadi *False* artinya *layer* tersebut tidak boleh dilatih. Pada *fine tuning* VGG16 ini, kami melatih block convolutional layer

Setiap *layer* memiliki parameter yang disebut *trainable*. Untuk membekukan bobot lapisan tertentu kita harus mengatur parameter ini ke *False*, menunjukkan bahwa *layer* tersebut tidak boleh dilatih. Pada VGG16, kami hanya melatih block block 3,4, dan 5 yang terdiri dari 12 *layer*.

```
14. for layer in prior.layers[:-12]:
15.     layer.trainable = False
16. for layer in prior.layers:
17.     print(layer, layer.trainable)
```

```

<tensorflow.python.keras.engine.input_layer.InputLayer object at
0x7ff1ff66d198> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff64c4a8> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff66d390> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at
0x7ff1ff64def0> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff64dd88> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff4e1f20860> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at
0x7ff1ff66dc18> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff67e550> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff656eb8> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff4e1e32b0> True
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at
0x7ff4e1e3ba8> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff6766d8> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff685550> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff685978> True
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at
0x7ff1ff68e550> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff66ed30> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff62bba8> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at
0x7ff1ff62bb38> True
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at
0x7ff1ff632898> True

```

Setelah menetapkan *layer trainable* pada model, kemudian *classifier* akan ditambahkan. *Layer fully-connected* yang kami buat dengan 4 output sesuai dengan jumlah kelas yang dimiliki.

```

18. model = keras.Sequential([
19.     keras.Input(shape=(224, 224, 3)),
20.     prior,
21.     keras.layers.Flatten(),
22.     keras.layers.Dense(1024, activation='relu'),
23.     keras.layers.Dropout(0.6),
24.     keras.layers.Dense(4, activation='softmax'),
25. ])
26. model.summary()

```

| Layer (type)        | Output Shape      | Param #  |
|---------------------|-------------------|----------|
| vgg16 (Functional)  | (None, 7, 7, 512) | 14714688 |
| Flatten_1 (Flatten) | (None, 25088)     | 0        |

|                     |              |          |
|---------------------|--------------|----------|
| dense_1 (Dense)     | (None, 1024) | 25691136 |
| dropout_1 (Dropout) | (None, 1024) | 0        |
| dense_2 (Dense)     | (None, 4)    | 4100     |

---

Total params: 40,409,924  
 Trainable params: 40,149,764  
 Non-trainable params: 260,160

---

#### 4.5.3 Implementasi Model Pretrained

Tahap selanjutnya adalah menguji perbandingan akurasi model pretrained dari Resnet50, InceptionV3, dan VGG16. Modul yang digunakan pada model pretrained sama dengan modul yang digunakan pada transfer learning yang berbeda hanyalah pada baris berikut.

```
for layer in base_model.layers:
    layer.trainable = True
model.summary()
```

Pada *trainable layer* parameter yang digunakan adalah *TRUE*, artinya kita akan melatih kembali semua model dari awal input hingga *fully*. Berikut adalah jumlah parameter yang akan penulis latih dengan model *transfer learning* InceptionNet.

---

Total params: 23,905,060  
 Trainable params: 23,870,628  
 Non-trainable params: 34,432

---

Sedangkan parameter yang digunakan pada model resnet adalah

---

Total params: 25,689,988  
 Trainable params: 25,636,868  
 Non-trainable params: 53,120

---

Dan parameter yang digunakan pada model VGG16 adalah

---

Total params: 15,244,100  
 Trainable params: 15,244,100  
 Non-trainable params: 0

---



#### 4.6. Tahap Implementasi Skenario Kedua

Skenario kedua adalah membandingkan hasil klasifikasi dari CNN *softmax* maka kita perlu melakukan pelatihan model terlebih dahulu. Parameter pelatihan dengan model CNN dijelaskan pada tabel 4.2. Pelatihan dimulai dengan meng-compile model. Kemudian menggunakan fungsi `model.fit()` untuk melakukan pelatihan menggunakan data latih dan data validasi sebagai pengukur kinerja model. `Model.evaluate()` digunakan untuk mendapatkan hasil dari pelatihan yang telah di uji dengan set uji yang belum diketahui oleh model saat pelatihan. Hasil inilah yang akan digunakan sebagai perbandingan akurasi klasifikasi dari model yang digunakan.

```

1. from tensorflow.keras import layers
2. from keras.callbacks import TensorBoard
3. from keras.optimizers import SGD
4. model.compile(optimizer=SGD(lr=0.0001, momentum=0.9, nesterov=False), loss='categorical_crossentropy', metrics=['accuracy'])
5. history = model.fit(datagen.flow(X_train, y_train), validation_data=(X_test, y_test), epochs=100, batch_size=82)
6. scores = model.evaluate(X_test, y_test, verbose=0)
7. print("Accuracy: %.2f%%" % (scores[1]*100))

```

Berikut adalah hasil dari proses pelatihan

```

Epoch 91/100
74/74 [-----] - 28s 378ms/step - loss: 0.0487 - accuracy: 0.9866 - val_loss: 0.0778 - val_accuracy: 0.9848
Epoch 92/100
74/74 [-----] - 28s 375ms/step - loss: 0.0453 - accuracy: 0.9840 - val_loss: 0.0700 - val_accuracy: 0.9848
Epoch 93/100
74/74 [-----] - 28s 378ms/step - loss: 0.0417 - accuracy: 0.9868 - val_loss: 0.0706 - val_accuracy: 0.9848
Epoch 94/100
74/74 [-----] - 28s 378ms/step - loss: 0.0469 - accuracy: 0.9832 - val_loss: 0.0725 - val_accuracy: 0.9848
Epoch 95/100
74/74 [-----] - 28s 378ms/step - loss: 0.0315 - accuracy: 0.9911 - val_loss: 0.0711 - val_accuracy: 0.9882
Epoch 96/100
74/74 [-----] - 28s 376ms/step - loss: 0.0320 - accuracy: 0.9908 - val_loss: 0.0724 - val_accuracy: 0.9848
Epoch 97/100
74/74 [-----] - 28s 377ms/step - loss: 0.0360 - accuracy: 0.9884 - val_loss: 0.0718 - val_accuracy: 0.9882

```

```

Epoch 98/100
74/74 [-----] - 28s 378ms/step - loss: 0.0300 -
accuracy: 0.9920 - val_loss: 0.0755 - val_accuracy: 0.9848
Epoch 99/100
74/74 [-----] - 28s 377ms/step - loss: 0.0303 -
accuracy: 0.9899 - val_loss: 0.0878 - val_accuracy: 0.9848
Epoch 100/100
74/74 [-----] - 28s 377ms/step - loss: 0.0335 -
accuracy: 0.9908 - val_loss: 0.0749 - val_accuracy: 0.9848

```

#### 4.7. Tahap Implementasi Skenario Ketiga

Skenario ketiga adalah melakukan perbandingan hasil klasifikasi menggunakan metode kombinasi CNN SVM. Untuk melakukan klasifikasi pada layer *fully* yang bawaan asli dari CNN juga bisa dilakukan klasifikasi dengan menggunakan metode yang lain. Misalnya saja dengan menggunakan SVM. Langkah yang dilakukan pada tahap ini adalah dengan melakukan ekstraksi fitur menggunakan CNN (konvolusi dan pooling). Setelah mendapatkan hasil ekstraksi fitur langkah selanjutnya adalah mengubah ukuran dimensi fitur dari konvolusi disesuaikan dengan dimensi pada SVM, maka kami melakukan reshape. Berikut adalah code untuk mengambil ekstraksi fitur dari CNN dan reshape dimensi input bagi SVM.

```

extractor = model.predict(X_train)
features = extractor.reshape(extractor.shape[0], -1)
X_SVM = features

```

Setelah mendapatkan input yang sesuai untuk SVM Langkah selanjutnya adalah klasifikasi menggunakan SVM. Sebelumnya kita import modul-modul yang diperlukan untuk klasifikasi SVM.

```

from sklearn.svm import LinearSVC
from sklearn import svm
from sklearn.svm import SVC

```

Model SVM yang kita gunakan adalah SVM linier karena kelas output yang kita harapkan berupa *multiclass classification*. Kode yang digunakan untuk inisialisasi SVM sebagai berikut

```
SVM model = svm.LinearSVC(
    C=50.0, dual=True, fit_intercept=True,
    intercept_scaling=1, loss='squared_hinge', max_iter=1000,
    multi_class='ovr', penalty='l2', tol=0.0001,
    verbose=0
)
```

Kemudian melakukan pelatihan pada data latih dengan kode sebagai berikut

```
SVM_model.fit(X_SVM,y_train)
```

Kemudian melakukan pengujian dengan data uji, terlebih dahulu kita melakukan reshape sebagai berikut

```
X_test_fiturs = model.predict(X_test)
X_test_fiturs = X_test_fiturs.reshape(X_test_fiturs.shape[0],-1)
prediction_SVM = SVM_model.predict(X_test_fiturs)
```

dan untuk mendapatkan nilai akurasi adalah

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, prediction_SVM)*100)
```

Untuk optimasi klasifikasi SVM kita dapat mencarinya dengan merubah nilai C. Nilai C akan memberikan pengaruh terhadap bentuk *hyperplane* serta hasil klasifikasi

#### 4.8. Tahap Implementasi Skenario Keempat

Untuk melakukan optimasi pada metode kombinasi CNN SVM, kami melakukan pengujian dengan melatih CNN-nya terlebih dahulu hingga mendapat hasil yang baik. Modelnya kami save, kemudian kami mengambil *fully connected layer*-nya dengan *code* seperti berikut.

```

model = Model(inputs=model.input, outputs=model.layers[-1].output)
model.summary()

```

Fitur yang didapat dari *fully connected layer* nantinya dibentuk kedalam *dataframe* untuk selanjutnya diproses ke *classifier machine learning* lain.

#### 4.9. Tahap Prediksi Klasifikasi Menggunakan *Confusion Matrix*

Pada tahap ini berisi perhitungan *confusion matrik* pada model klasifikasi CNN menggunakan *code* sebagai berikut.

```

1. y_pred = model.predict(X_test)
# ITERASI X TEST
2. y_prediction_n = []
3. y_test_n = []
4. for i in range(0, len(y_pred)):
5.     max_n = max(y_pred[i])
6.     max_m = max(y_test[i])
7.     index_n = [index_n for index_n, j in enumerate(y_pred[i]) if j == max_n]
8.     index_m = [index_m for index_m, j in enumerate(y_test[i]) if j == max_m]
9.     y_prediction_n.append(index_n)
10.    y_test_n.append(index_m)
#CLASSIFICATION REPORT
11. from sklearn.metrics import classification_report
12. print(classification_report(y_test_n, y_prediction_n, target_names=["Severe",
    Mild", "Moderate", "NORMAL"]))
13. y_pred = model.predict_classes(X_test)
14. y_pred
15. y_test_class = np.argmax(y_test, axis=1)
16. from sklearn.metrics import confusion_matrix, accuracy_score
17. import seaborn as sns
18. print(accuracy_score(y_test_class, y_pred))
19. confusion_matrix(y_test_class, y_pred)

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Severe       | 0.81      | 1.00   | 0.90     | 47      |
| Mild         | 1.00      | 0.79   | 0.88     | 52      |
| Moderate     | 0.96      | 0.98   | 0.97     | 130     |
| NORMAL       | 1.00      | 0.97   | 0.98     | 67      |
| accuracy     |           |        | 0.95     | 296     |
| macro avg    | 0.94      | 0.93   | 0.93     | 296     |
| weighted avg | 0.95      | 0.95   | 0.95     | 296     |

Sedangkan pada tahap klasifikasi menggunakan SVM menggunakan kode sebagai berikut.

```

1. X_test_fiture = modelTL.predict(X_test)
2. X_test_fitures = X_test_fiture.reshape(X_test_fiture.shape[0],-1)
3. prediction_SVM = SVM_model.predict(X_test_fitures)
4. prediction_SVM = le.inverse_transform(prediction_SVM)
5. from sklearn import metrics
6. print("Accuracy=", metrics.accuracy_score(y_test, prediction_SVM))
7. #CLASSIFICATION REPORT
8. from sklearn.metrics import classification_report
9. print(classification_report(y_test, prediction_SVM, target_names=["Severe", "Mi
    ld", "Moderate", "NORMAL"]))

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Severe       | 0.78      | 0.77   | 0.77     | 47      |
| Mild         | 0.71      | 0.71   | 0.71     | 52      |
| Moderate     | 0.82      | 0.84   | 0.83     | 130     |
| NORMAL       | 0.77      | 0.75   | 0.76     | 67      |
| accuracy     |           |        | 0.78     | 296     |
| macro avg    | 0.77      | 0.77   | 0.77     | 296     |
| weighted avg | 0.78      | 0.78   | 0.78     | 296     |

#### 4.10. Analisis dan Pembahasan

##### 4.10.1 Konfigurasi Penelitian

Setiap skenario pengujian menggunakan *preprocessing green channel*, CLAHE, *resize* gambar dengan ukuran (224 x 224) px, dan menggunakan data augmentasi. Skenario pengujian akan kami definisikan dengan parameter CNN yang kami gunakan ditunjukkan pada tabel 4. 5.

Tabel 4. 5. Parameter Pelatihan menggunakan CNN

| Optimizer     | SGD                      |
|---------------|--------------------------|
| Learning Rate | menyesuaikan             |
| Loss          | categorical_crossentropy |
| Metric        | accuracy                 |
| Epoch         | menyesuaikan             |
| Batch Size    | 82                       |
| Dropout       | 0.6                      |

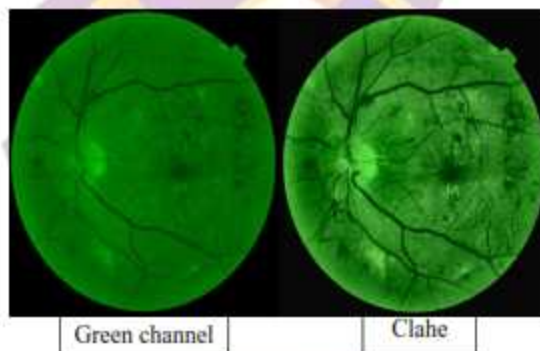
*Epoch* adalah keadaan ketika dataset mengunjungi semua proses jaringan neural dari awal hingga akhir dan kembali lagi ke nilai awal untuk satu siklus

(*backpropagation*). Learning rate adalah ukuran Langkah pada setiap iterasi yang dilakukan.

Sedangkan parameter SVM yang digunakan adalah SVM *linear* dengan nilai  $C = 50$ , dan nilai *penalty* = 11.

#### 4.10.2 Hasil Implementasi Preprocessing

Preprocessing yang digunakan adalah mengubah gambar RGB untuk hanya diambil *channel* hijau saja yang kemudian menggunakan CLAHE untuk meningkatkan kontras hasil dari *green channel* dan CLAHE dapat dilihat pada gambar 4. 2 sebagai berikut.

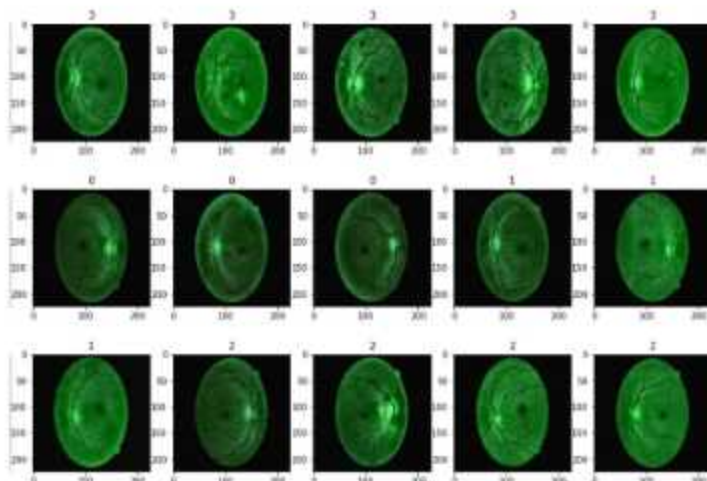


Gambar 4. 2. Hasil *Green Channel* dan CLAHE

#### 4.10.3 Hasil Implementasi Data Augmentasi

a. Hasil *Horizontal Flip*

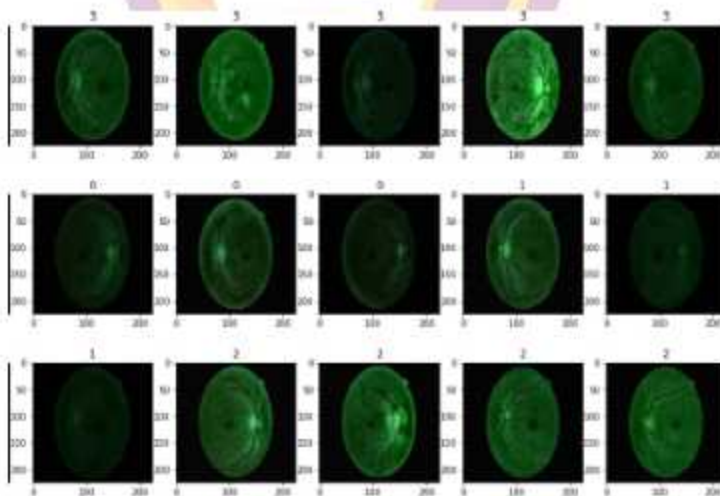
Berikut adalah hasil implementasi dari augmentasi data *horizontal flip* ditunjukkan pada gambar 4. 3.



Gambar 4. 3. Hasil dari Augmentasi *Horizontal Flip*

b. Hasil *Random Brightness*

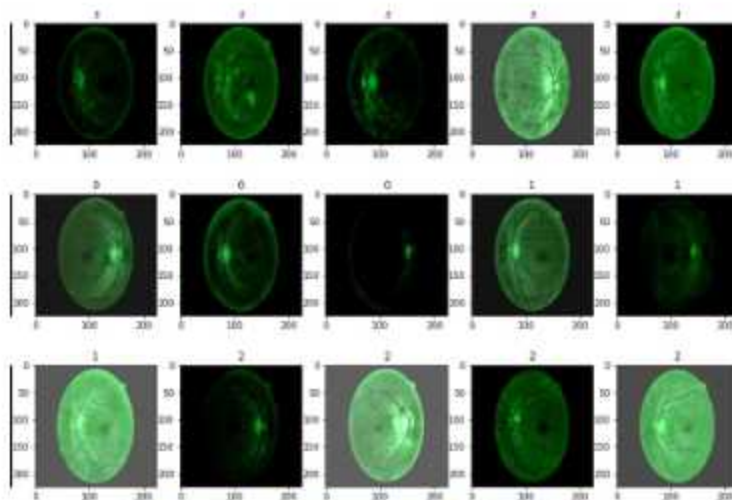
Berikut adalah hasil implementasi dari *random brightness* ditunjukkan pada gambar 4. 4.



Gambar 4. 4. Hasil dari Augmentasi *Random Brightness*

c. Hasil *Channel Shift Range*

Berikut adalah hasil implementasi dari *channel shift range* ditunjukkan pada gambar 4. 5.



Gambar 4. 5. Hasil dari Augmentasi *Channel Shift Range*

#### 4.10.4 Hasil Implementasi Arsitektur Transfer Learning pada Klasifikasi menggunakan metode CNN

Untuk hasil percobaan menggunakan model CNN *transfer learning* murni, dengan memanfaatkan jaringan terlatih dari VGG16, InceptionNet dan Resnet50 dengan hanya mengubah pada *layer fully* menyesuaikan dengan klasifikasi data yang penulis miliki, pada klasifikasi menggunakan CNN. Penulis mendapatkan hasil sebagaimana pada tabel 4.6.

Tabel 4. 6. Hasil Implementasi *Transfer Learning* Klasifikasi pada CNN *Softmax*

| <i>Transfer learning</i> | Akurasi yang diperoleh (%) |
|--------------------------|----------------------------|
| Inception V3             | 53,72                      |
| Resnet50                 | 38,96                      |
| Vgg16                    | 40,77                      |



Dari tabel 4. 6 kita dapatkan bahwa pengujian dengan model *transfer learning* VGG16, Inception V3, dan Resnet50, mendapatkan akurasi yang buruk. Hal itu dikarenakan, walaupun pembelajaran *transfer* dari dataset *ImageNet* meningkatkan kinerja dari tugas pengenalan pola namun pada dataset citra medis pembelajaran *transfer* tidak meningkatkan pola secara signifikan. Hal itu disebabkan karena dataset citra medis sangat berbeda dengan kumpulan data pada jaringan yang telah dilatih sebelumnya. Jika ingin meningkatkan hasil lebih baik lagi maka kita bisa menggunakan *fine tuning*.

#### 4.10.5 Hasil Implementasi Arsitektur Transfer Learning pada Klasifikasi Menggunakan Metode Kombinasi CNN SVM

Berikut adalah hasil pengujian kombinasi CNN dan SVM. Dimana CNN berfungsi sebagai ekstraksi fitur. Fitur yang didapat dari ekstraksi fitur layer CNN nantinya dibentuk kedalam *dataframe* untuk selanjutnya diproses ke *classifier machine learning* SVM. Berikut adalah ekstraksi fitur dari masing-masing model. Diterangkan pada gambar 4.6 adalah ekstraksi fitur dari model VGG16, gambar 4.7 ekstraksi fitur dari model Resnet50, dan gambar 4.8 adalah ekstraksi fitur dari model InceptionV3.

|      | feature001 | feature002 | feature003 | feature004 | feature005 | feature007 | feature008 | feature009 | feature010 | feature011 | label |     |
|------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------|-----|
| 0    | 0.701604   | 0.329522   | -0.300080  | 0.000000   | 0.340741   | 0.000000   | 1.348317   | 0.0        | 0.000000   | 1.477432   | 0.0   | 2   |
| 1    | 0.200000   | 1.000022   | -0.200000  | 0.207579   | 0.400339   | 0.201338   | 0.245404   | 0.0        | 0.013953   | 0.152791   | 0.0   | 1   |
| 2    | 0.400003   | 1.100042   | 0.300029   | 0.511730   | 0.100425   | 0.042339   | 1.101335   | 0.0        | 0.000000   | 2.136305   | 0.0   | 2   |
| 3    | 0.079802   | 1.020002   | 0.121481   | 0.416723   | 1.412294   | 0.257747   | 0.097058   | 0.0        | 0.000000   | 1.417212   | 0.0   | 2   |
| 4    | 0.081082   | 1.040070   | 0.101284   | 1.431138   | 0.202473   | 0.183325   | 0.000360   | 0.0        | 0.000000   | 0.820071   | 0.0   | 0   |
| ...  | ...        | ...        | ...        | ...        | ...        | ...        | ...        | ...        | ...        | ...        | ...   | ... |
| 2061 | 0.123261   | 0.808862   | 2.150088   | 0.321082   | 1.102512   | 0.616166   | 0.709758   | 0.0        | 0.000000   | 0.820071   | 0.0   | 2   |
| 2062 | 0.041021   | 1.060511   | 0.187881   | 0.130528   | 0.003922   | 0.011591   | 0.644335   | 0.0        | 0.000000   | 1.401587   | 0.0   | 2   |
| 2063 | 0.080008   | 1.050007   | 0.300001   | 0.316616   | 0.004429   | 0.018575   | 1.432340   | 0.0        | 0.000000   | 1.401001   | 0.0   | 2   |
| 2064 | 0.240071   | 1.010002   | 0.300070   | 0.407720   | 0.016115   | 0.018745   | 0.137682   | 0.0        | 0.000122   | 0.040020   | 0.0   | 1   |
| 2065 | 0.020442   | 1.410027   | 0.170020   | 0.000643   | 0.016932   | 0.288791   | 1.282378   | 0.0        | 0.014249   | 1.191676   | 0.0   | 2   |

2066 (yml) = 2046 (column)

Gambar 4. 6. Ekstraksi Fitur dari VGG16

|      | feature0 | feature1 | feature2 | feature3 | feature4 | feature5 | feature6 | feature7 | feature8 | feature9 | feature10 | feature11 | label |
|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-------|
| 0    | 0.48322  | 0.42039  | 0.00000  | 0.09489  | -0.20317 | 0.21381  | 0.01647  | 0.91378  | -0.79000 | 0.22388  | 0.08701   | 0.09788   | 3     |
| 1    | 0.29433  | 0.42000  | 0.00000  | 0.21699  | 0.01591  | 0.27000  | 0.40415  | 0.39989  | -0.28754 | 0.47991  | 0.23042   | 0.34814   | 2     |
| 2    | 0.62771  | 0.74204  | 0.00000  | 0.00000  | 0.49732  | 0.22509  | 0.20327  | 0.13088  | 0.49422  | -0.23019 | 0.37022   | 0.20471   | 2     |
| 3    | 0.20903  | 0.79780  | 0.40107  | 0.39422  | 0.81430  | 0.00077  | 0.19042  | 0.37662  | 0.07343  | 0.00022  | 0.01288   | 0.02868   | 3     |
| 4    | 0.87042  | 0.76363  | 0.74517  | 0.31825  | -0.89435 | 0.43703  | 0.88103  | 0.12472  | 0.30498  | 0.04228  | 0.06664   | 0.02152   | 3     |
| ...  | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...       | ...       | ...   |
| 2091 | 0.18024  | 0.44347  | 0.11219  | 0.29009  | -0.29109 | 0.33042  | 0.70041  | 0.01291  | 0.26410  | 0.14079  | 0.06479   | 0.09736   | 3     |
| 2092 | 0.21637  | 0.08003  | 0.18645  | 0.11949  | 0.03468  | 0.14709  | 0.32044  | 0.07911  | 0.17272  | 0.32287  | -0.21081  | 0.21419   | 1     |
| 2093 | 0.28049  | 0.07721  | 0.02352  | 0.10488  | 0.20730  | 0.40005  | 0.00077  | 0.34688  | 0.00871  | 0.00000  | 0.13047   | 0.00000   | 2     |
| 2094 | 0.08003  | 0.02000  | 0.00000  | 0.12000  | 0.00000  | 0.20000  | 0.00000  | 0.01000  | 0.00000  | 0.00000  | 0.00000   | 0.00000   | 2     |
| 2095 | 0.30755  | 0.25912  | 0.25091  | 0.03072  | 0.25072  | 0.40033  | 0.00712  | 0.14047  | 0.00011  | 0.22728  | 0.01075   | 0.00011   | 2     |

2096 rows x 2018 columns

Gambar 4. 7. Ekstraksi Fitur Resne50

|      | feature0 | feature1 | feature2 | feature3 | feature4 | feature5 | feature6 | feature7 | feature8 | feature9 | feature10 | feature11 | label |
|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-------|
| 0    | 0.26001  | 0.42012  | 0.32342  | 0.24012  | 0.12018  | 0.14018  | 0.00012  | 0.01015  | 0.01198  | 0.01423  | 0.00020   | 0.00020   | 3     |
| 1    | 0.47071  | 0.38703  | 0.19508  | 0.44907  | 0.01540  | 0.00781  | 0.01786  | 0.28712  | 0.28926  | 0.07225  | 0.26019   | 0.00000   | 1     |
| 2    | 0.15071  | 0.00003  | 0.94722  | 0.10000  | 0.00000  | 0.00000  | 0.40764  | 0.27010  | 0.00043  | -0.21529 | 0.15481   | 0.00000   | 1     |
| 3    | 0.17423  | 0.21002  | 0.20244  | 0.22200  | 0.22072  | 0.20113  | 0.00035  | 0.20542  | 0.00000  | 0.65420  | 0.00000   | 0.00000   | 2     |
| 4    | 0.21000  | 0.21000  | 0.21000  | 0.21000  | 0.21000  | 0.21000  | 0.20072  | 0.20000  | 0.07015  | 0.10000  | 0.20072   | 0.00000   | 3     |
| ...  | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...      | ...       | ...       | ...   |
| 2091 | 0.20000  | 0.01145  | 0.00000  | 0.22000  | 0.00000  | 0.00000  | 0.10000  | 0.20000  | 0.14000  | 0.01000  | 0.00000   | 0.00000   | 3     |
| 2092 | 0.17001  | 0.21000  | 0.01000  | 0.17000  | 0.40000  | 0.25000  | 0.17000  | 0.27000  | 0.00000  | 0.00000  | 0.00000   | 0.00000   | 3     |
| 2093 | 0.10000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000   | 0.00000   | 2     |
| 2094 | 0.00000  | 0.00000  | 0.00000  | 0.20000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000   | 0.00000   | 2     |
| 2095 | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000  | 0.00000   | 0.00000   | 2     |

2096 rows x 2018 columns

Gambar 4. 8. Ekstraksi Fitur InceptionV3

Ekstraksi fitur pada klasifikasi ini diambil langsung dari model *transfer learning* yang tersedia pada *tensorflow keras* menggunakan model Resnet50, InceptionV3, dan VGG16. Hasil dari percobaan ini ditunjukkan pada tabel 4. 7.

Tabel 4. 7. Hasil Perbandingan Model Kombinasi CNN SVM

| Model kombinasi CNN dan SVM | Akurasi (%) |
|-----------------------------|-------------|
| VGG16                       | 79,72       |
| Inception V3                | 79,45       |
| ResNet50                    | 76,52       |

Berdasarkan pada tabel 4. 7 kita dapatkan hasil terbaik adalah penggunaan model VGG16 kombinasi SVM sebesar 79,72%. Pada penggunaan arsitektur *transfer learning* untuk klasifikasi metode kombinasi CNN SVM memberikan hasil yang jauh lebih baik daripada CNN softmax. Namun dalam penelitian ini, penulis

belum merasa puas dengan hasil yang diperoleh, oleh sebab itu penulis melakukan pengujian lanjutan sesuai dengan skenario yang telah penulis buat sebelumnya.

#### 4.10.6 Hasil Implementasi Model Fine Tuning VGG16

Berdasarkan tabel 4. 7 sebelumnya, dikarenakan hasil akurasi *transfer learning* VGG16 kurang bagus maka penulis mencoba melakukan *fine tuning model* VGG16, dengan harapan akan mendapatkan peningkatan akurasi menjadi lebih baik. Kelebihan VGG16 adalah memiliki desain arsitektur yang sederhana. Karena itu penulis mencoba melakukan *fine tuning* pada setiap *block layer* pada VGG16 dan mendapatkan hasil yang dapat dilihat pada tabel 4. 8.

Tabel 4. 8. Hasil *Fine Tuning*

| Nama block yang di latih | Akurasi yang diperoleh (%) |
|--------------------------|----------------------------|
| Layer Block 5            | 88,18                      |
| Layer Block 4 dan 5      | 89,53                      |
| Layer Block 3,4 dan 5    | 86,82                      |
| Layer Block 2,3,4 dan 5  | 87,16                      |

Dari tabel 4. 8 dapat kita lihat bahwa model dapat memberikan akurasi dengan baik jika kita melatih ulang layer pada blok 4 dan 5. Dan hasil tidak serta merta selalu menjadi semakin besar jika blok layer yang di latih semakin banyak. Bagaimana jika kita menggunakan semua layer untuk kita latih kembali (*pretrained model*) apakah nilai akurasi yang diberikan jauh lebih bagus? Maka penulis melakukan pengujian pada tahap selanjutnya untuk mendapatkan hasil sesuai yang diharapkan.

#### 4.10.7 Hasil Implementasi Klasifikasi menggunakan Model Pretrained CNN

Berdasarkan pada hasil percobaan pada tabel 4. 7 dan tabel 4. 8 maka untuk percobaan selanjutnya penulis akan membandingkan performa dari model

*pretrained* VGG16, InceptionNet V3, dan ResNet50. Dan hasilnya dapat dilihat pada tabel 4. 9 sebagai berikut.

Tabel 4. 9. Tabel Perbandingan *Pretrained Model*

| <i>Pretrained Model</i> | Akurasi (%)  | Sensitivitas (%) |
|-------------------------|--------------|------------------|
| VGG16                   | 90,77        | 91,53            |
| InceptionNet v3         | <b>97,29</b> | <b>97,49</b>     |
| Resnet50                | 96,16        | 95,76            |

Berdasarkan pada tabel 4. 9 tersebut nilai akurasi yang diperoleh dari setiap *pretrained* model adalah sangat bagus antara satu model dengan model lainnya. Karena itu untuk mendapatkan model terbaik parameter penilaian penulis tambahkan nilai sensitifitas. Karena sensitifitas pada analisis medis sangat penting. Sensitifitas memprediksi bahwa tidak ada pasien yang menderita DR diprediksi tidak menderita DR. Berdasarkan pada tabel 4. 9 kita dapatkan model dengan akurasi dan sensitivitas tertinggi adalah model Inception softmax dengan akurasi 97,29% dan sensitivitas 97,49%.

#### **4.10.8 Perbandingan Rasio Data (*Data Split*)**

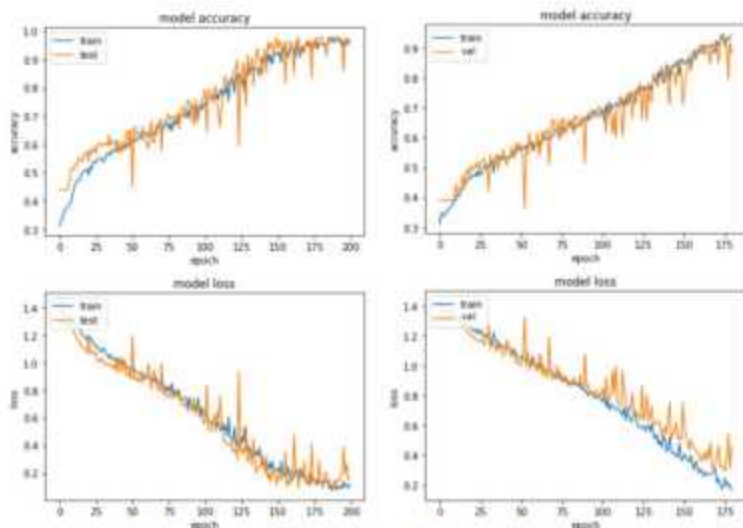
Seperti yang telah sudah dibahas pada bab sebelumnya bahwa *deep learning* sangat rentang terjadi *overfitting*. Berbagai hal telah dilakukan untuk mencegah *overfitting* seperti dengan menggunakan data augmentasi (Agustin et al., 2020), menambahkan *regularization* (Agustin & Sunyoto, 2020), menambah jumlah data, atau dengan menggunakan rasio perbandingan data seperti yang dijelaskan pada bab 2. Pada subbab ini penulis akan menginformasikan temuan yang telah penulis dapatkan setelah melakukan pengujian dengan membagi rasio data. Berikut adalah hasil perbandingan rasio data yang digunakan, yaitu dua *data split* yang terdiri dari data latih dan data uji dengan perbandingan 90:10 dan 3 *data split* yang terdiri dari

data latih, data validasi, dan data uji dengan perbandingan 70:15:15. Hasil dari pengujian penulis tuang dalam tabel 4. 10 agar lebih untuk dipahami.

Tabel 4. 10. Hasil Perbandingan Rasio Data

| Model               | Akurasi (%)  |              |
|---------------------|--------------|--------------|
|                     | 2 split data | 3 split data |
| VGG16 softmax       | 96,96        | 90,77        |
| InceptionV3 softmax | 99,32        | 97,29        |
| Resnet50 softmax    | 98,99        | 96,16        |

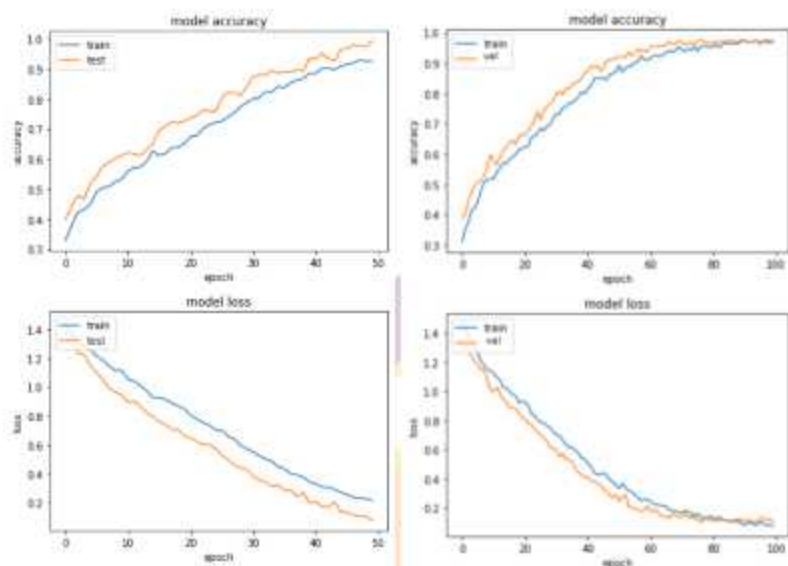
Dari hasil tabel 4. 10 tersebut, kita dapatkan bahwa nilai akurasi lebih tinggi kita dapatkan dengan mengelompokkan data ke dalam data latih dan data uji saja. Dengan semakin besar porsi data latih akan meningkatkan akurasi menjadi lebih baik. Namun dalam menilai performa CNN tidak bisa hanya dengan berpatokan pada hasil akurasi tinggi yang diperoleh namun juga dengan menilai performa jaringan, apakah jaringan yang telah dibuat menunjukkan *good fit* dan bukan *overfit*. Oleh sebab itu kita bisa membandingkan performa jaringan dengan menggunakan grafik *fit model* untuk menilai performa jaringan. Berikut adalah grafik *fit model* yang diperoleh oleh masing-masing model yang merupakan hasil dari pelatihan data latih, dan data tes pada 2 *split* data atau hasil dari pelatihan data latih, data validasi, dan data tes pada 3 *split* data. Grafik tersebut menunjukkan pergerakan nilai akurasi (*accuracy*) dan nilai loss (*loss*) untuk data latih dan data validasi yang dihasilkan pada setiap iterasi (*epoch*). Dari grafik ini dapat diketahui bahwa iterasi menghasilkan nilai akurasi dan nilai *loss* untuk data latih dan data validasi. Nilai akurasi adalah suatu nilai yang digunakan untuk mengetahui tingkat keberhasilan model yang telah dibuat. Sedangkan nilai *loss* merupakan suatu ukuran dari sebuah *error* yang dibuat oleh jaringan, dan tujuannya adalah untuk meminimalisirnya.



Gambar 4. 9. Grafik Fit Model Perbandingan 2 Split Data Dengan 3 Split Data Pada Model VGG 16

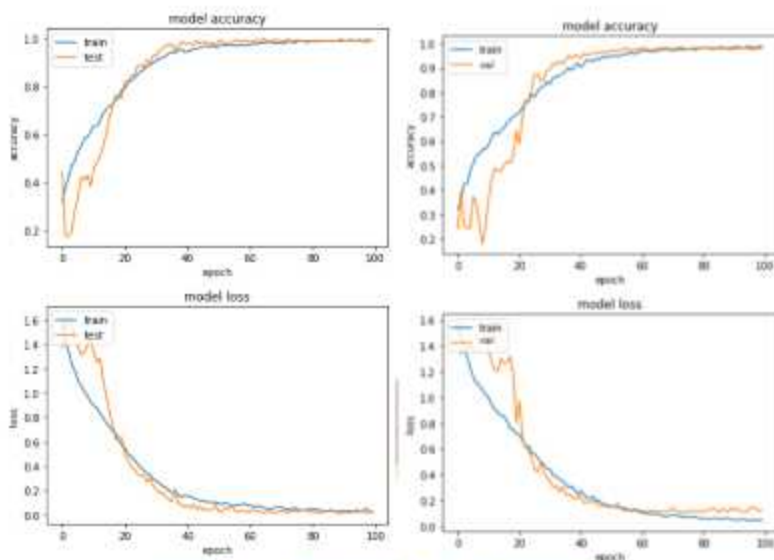
Grafik *fit model* VGG16 pada gambar 4. 9 dilatih pada epoch mencapai 200. Grafik bagian atas menunjukkan nilai akurasi dari kedua data. Berdasarkan gambar 4. 9 dapat dilihat bahwa nilai akurasi hasil iterasi yang dihasilkan baik untuk data latih naik hingga *epoch* ke-200, sedangkan untuk data test mengalami naik turun sampai ke *epoch* ke-200. Sedangkan grafik bawah menunjukkan nilai *loss* untuk kedua data, dapat dilihat bahwa untuk data train, nilai *loss* yang didapatkan terus turun hingga *epoch* ke-200, dan nilai *loss* untuk data test naik turun dengan konsisten semakin menurun hingga pada sebuah titik nilai *loss* semakin naik hingga *epoch* ke-200. Berdasarkan pada gambar 4. 9 dapat dilihat bahwa performa model sudah bagus pada 2 *split* data. Berdasarkan grafik *loss* pada 3 *split* data, garis merah menunjukkan fungsi kerugian pada setiap *epoch* lebih tinggi daripada hasil

pembelajaran dari data latih (garis biru) menunjukkan terjadinya *overfitting*. Idealnya semakin tinggi *epoch*, fungsi kerugian akan semakin kecil.



Gambar 4. 10. Grafik Fit Model Perbandingan 2 Split Data Dengan 3 Split Data Pada Model Inceptionv3

Berdasarkan pada grafik iterasi pada model Inception V3 seperti pada gambar 4. 10 menggunakan iterasi (*epoch*) sebanyak 100. Nilai akurasi hasil iterasi yang dihasilkan untuk data latih naik hingga *epoch* ke-100, begitu juga nilai akurasi hasil iterasi yang dihasilkan dari data validasi semakin naik beriringan dengan data latih hingga *epoch* 100. Demikian juga dengan nilai loss pada data latih dan data validasi semakin menurun secara beriringan menunjukkan bahwa performa jaringan menunjukkan *good fit* pada 2 *split* data maupun pada 3 *split* data.



Gambar 4. 11. Grafik Fit Model Perbandingan 2 Split Data Dengan 3 Split Data Pada Model Resnet 50

Berdasarkan pada grafik iterasi pada model Resnet50 seperti pada gambar 4. 11 menggunakan iterasi (*epoch*) sebanyak 100. Nilai akurasi hasil iterasi yang dihasilkan untuk data latih naik hingga *epoch* ke-100, begitu juga nilai akurasi hasil iterasi yang dihasilkan dari data validasi semakin naik beriringan dengan data latih hingga *epoch* 100. Demikian juga dengan nilai loss pada data latih dan data validasi semakin menurun secara beriringan walaupun pada grafik 3 split data nilai loss data validasi mengalami sedikit peningkatan mencapai 0,1205 di atas nilai loss data latih pada 0.0574. Namun garifik tersebut masih menunjukkan bahwa performa jaringan menunjukkan *good fit* pada 2 *split* data maupun pada 3 *split* data.



#### 4.10.9 Hasil Implementasi Optimalisasi Model Kombinasi CNN SVM

Langkah yang dilakukan pada skenario empat ini adalah data di ekstraksi menggunakan arsitektur model Resnet50, InceptionV3, dan VGG16 yang dilatih kembali (*pretrained*). Hasil dari model terlatih ini kemudian disimpan kedalam file model.sh sesuai dengan masing-masing arsitekturnya. Kemudian untuk klasifikasi dengan SVM model ini di *load* (dipanggil) dengan hanya mengambil lapisan ekstraksi fitur saja, pada bagian *fully layer* digantikan menggunakan klasifikasi *mechine learning* SVM. Hasil dari pengujian ini ditunjukkan pada tabel 4. 11 sebagai berikut.

Tabel 4. 11. Hasil Perbandingan Model Kombinasi optimalisasi CNN SVM

| <i>Pretrained Model</i> | Akurasi (%)  | Sensitivitas (%) |
|-------------------------|--------------|------------------|
| VGG16                   | 98,42        | 97,98            |
| InceptionNet v3         | 99,09        | 98,96            |
| Resnet50                | <b>99,77</b> | <b>99,63</b>     |

Berdasarkan pada tabel 4. 11 tersebut nilai akurasi yang diperoleh dari setiap *pretrained* model ekstraksi fitur kombinasi SVM adalah sangat bagus antara satu model dengan model lainnya. Karena itu untuk mendapatkan model terbaik parameter penilaian kami tambahkan nilai sensitifitas. Berdasarkan pada tabel 4. 11 kita dapatkan model dengan akurasi dan sensitivitas tertinggi adalah model Resnet SVM dengan akurasi sebesar 99,77% dan sensitivitas 99,63%.

#### 4.10.10 Perbandingan Hasil Klasifikasi CNN dan kombinasi CNN dan SVM

Berikut adalah tabel perbandingan semua metode yang digunakan ditunjukkan pada tabel 4. 12 dibawah ini.

Tabel 4. 12. Perbandingan Hasil Model CNN Softmax dan CNN SVM

| Model               | Akurasi (%)  | Sensitivitas (%) |
|---------------------|--------------|------------------|
| VGG16 softmax       | 90,77        | 91,53            |
| InceptionV3 softmax | 97,29        | 97,49            |
| Resnet50 softmax    | 96,16        | 95,76            |
| VGG16 SVM           | 98,42        | 97,98            |
| InceptionV3 SVM     | 99,09        | 98,96            |
| ResNet50 SVM        | <b>99,77</b> | <b>99,63</b>     |

Berdasarkan tabel 4. 12 kita peroleh hasil dari penelitian ini adalah metode kombinasi klasifikasi CNN SVM menghasilkan akurasi dan sensitivitas yang tinggi mengungguli metode klasifikasi CNN softmax pada klasifikasi multi kelas dengan nilai akurasi yang diperoleh sebesar 99,77% dan nilai sensitivitas sebesar 99,63% dengan menggunakan model ekstraksi fitur pretrained Resnet50. Penelitian ini menunjukkan bahwa model transfer learning yang di latih kembali (*pretrained*) dapat memberikan nilai akurasi yang tinggi.

#### 4.10.11 Grafik Perbandingan Hasil Klasifikasi CNN dan kombinasi CNN dan SVM



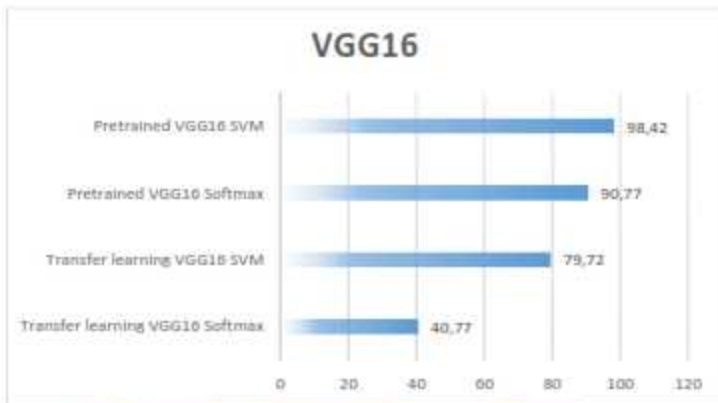
Gambar 4. 12. Perbandingan Klasifikasi Metode CNN dengan Metode Kombinasi CNN dan SVM

Dari grafik pada gambar 4. 12 tersebut kita dapatkan bahwa nilai akurasi dan sensitifitas pada klasifikasi menggunakan *softmax* mendapatkan akurasi yang tinggi pada model InceptionV3. Sedangkan pada klasifikasi dengan SVM nilai akurasi dan sensitivitas tertinggi kita peroleh pada model Resnet50. Model VGG16 menghasilkan akurasi dan sensitivitas lebih rendah dibandingkan kedua model lainnya. Walaupun model arsitektur VGG16 lebih sederhana namun model ini memiliki waktu pelatihan yang lebih lama, dan menghasilkan akurasi yang lebih rendah. Selain itu informasi yang kita dapatkan berdasarkan tabel di atas adalah bahwa klasifikasi menggunakan SVM meningkatkan hasil akurasi hingga mencapai 3% lebih dibandingkan klasifikasi menggunakan *softmax* CNN pada arsitektur Resnet50. Pada model InceptionV3 peningkatan akurasi yang diperoleh adalah sebesar 1,8%. Sedangkan lebih besar lagi peningkatan akurasi pada model VGG16 yang mencapai lebih dari 7%.

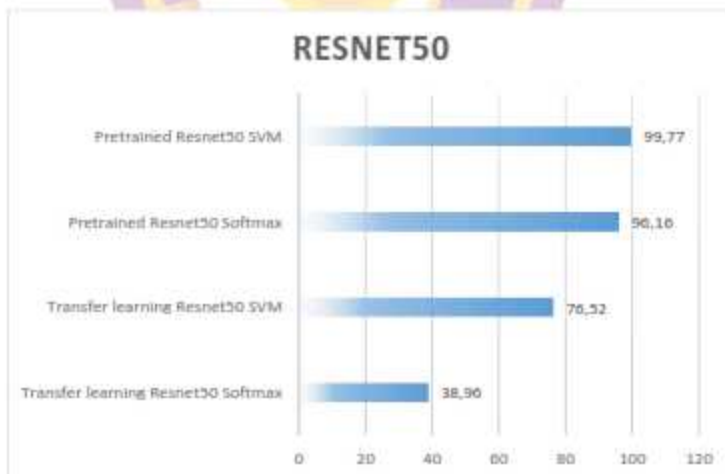
#### **4.10.12 Grafik Perbandingan Hasil Klasifikasi Peningkatan Akurasi yang Diperoleh dari Evaluasi Awal hingga Akhir**

Berikut adalah penjelasan peningkatan akurasi yang diperoleh dari setiap skenario pengujian perbandingan klasifikasi CNN *Softmax* dan CNN SVM pada setiap model arsitektur CNN yang digunakan. Berdasarkan grafik pada gambar 4.13, 4.14, dan 4.15 kita dapatkan bahwa setiap klasifikasi menggunakan metode kombinasi CNN SVM memberikan hasil akurasi yang lebih tinggi dibandingkan dengan metode klasifikasi CNN *Softmax*, peningkatan yang diperoleh pun cukup signifikan meningkatkan hasil akurasi. Dengan menggunakan metode optimalisasi kombinasi CNN SVM yaitu dengan melatih kembali (*pretrained*) model *transfer*

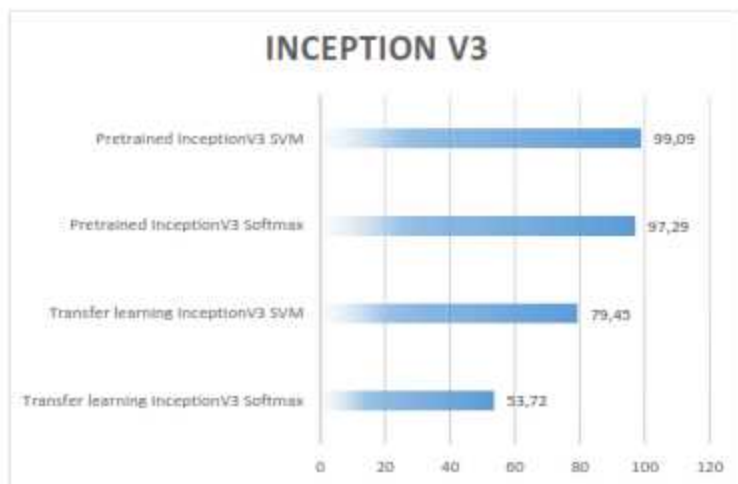
*learning* maka semakin meningkatkan hasil akurasi yang diperoleh pada setiap model arsitektur yang digunakan. Dan akhirnya kita memperoleh bahwa model arsitektur Resnet50 dengan klasifikasi SVM memberikan nilai akurasi tertinggi dari pada kedua model lainnya.



Gambar 4. 13 Peningkatan akurasi pada model VGG16



Gambar 4. 14. Peningkatan akurasi pada model Resnet50



Gambar 4. 15. Peningkatan akurasi pada model InceptionV3

#### 4.10.13 Hasil perhitungan *Confusion Matrix*

Perhitungan *confusion matrix* yang kami ambil adalah dari hasil uji coba dengan nilai akurasi terbaik yaitu dari metode kombinasi CNN SVM dengan model arsitektur Resnet50. Tabel 4.13 adalah grafik *confusion matrix* yang kami peroleh.

Tabel 4. 13. Grafik Confusion Matrix

|                           |   | Data Prediksi |    |     |     | Total |
|---------------------------|---|---------------|----|-----|-----|-------|
|                           |   | 0             | 1  | 2   | 3   |       |
| Data Benar Level Penyakit | 0 | 169           | 1  | 0   | 0   | 169   |
|                           | 1 | 0             | 67 | 0   | 0   | 68    |
|                           | 2 | 0             | 0  | 100 | 0   | 100   |
|                           | 3 | 0             | 0  | 0   | 106 | 106   |
| Total                     |   | 170           | 67 | 100 | 106 | 443   |

Untuk menghitung nilai masing-masing dari hasil *confusion matrix* digunakan persamaan 1,2,3, dan 4 yang telah dijelaskan pada bab 2 sebelumnya. Maka hasilnya akan kami simulasikan kedalam tabel 4.14 sebagai berikut.

Tabel 4. 14. Perhitungan Hasil Confusion Matrix

|              | 0          | 1        | 2   | 3   | Hasil (%) |
|--------------|------------|----------|-----|-----|-----------|
| TP           | 169        | 67       | 100 | 106 | -         |
| TN           | 274        | 376      | 343 | 337 | -         |
| FP           | 0          | 1        | 0   | 0   | -         |
| FN           | 1          | 0        | 0   | 0   | -         |
| Akurasi      | -          | -        | -   | -   | 99,77427  |
| sensitivitas | 100        | 98,52941 | 100 | 100 | 99,63235  |
| spesifitas   | 99,6363636 | 100      | 100 | 100 | 99,90909  |

Berdasarkan pada tabel 4.14 prediksi FN sebesar 1 ada pada prediksi mata normal sedangkan prediksi untuk mata DR bernilai 0 pada FN. Hal ini menunjukkan bahwa performa model yang kami buat sangat baik. FN (nilai prediksi negatif dikenal sebagai *recall* atau sensitifitas) artinya pasien sebenarnya menderita DR namun diprediksi tidak menderita DR. Jika ada pasien DR diprediksi tidak menderita DR akan berbahaya karena pasien luput dari perawatan dan akan menyebabkan perburukan penyakit. Model kami memprediksi bahwa tidak ada pasien yang menderita DR diprediksi tidak menderita DR. FP (nilai prediksi positif dikenal sebagai *presisi*) artinya pasien yang tidak menderita DR diprediksi menderita DR, untuk pasien ini pada pemeriksaan tahap selanjutnya dapat diketahui bahwa sebenarnya dia sehat. Oleh sebab itu penelitian ini tidak menggunakan presisi sebagai ukuran penilaian kelayakan system.

Pengujian pada penelitian didasarkan pada akurasi karena akurasi digunakan untuk mengukur model apakah dapat bekerja dengan baik untuk membedakan retina yang sehat dari retina penderita NPDR. Sensitivitas digunakan untuk mengukur hasil tes yang bernilai harus negatif jika digunakan untuk orang sehat. Sedangkan spesifitas menunjukkan kemampuan tes untuk memastikan retina

yang sehat dari retina yang benar-benar sehat. Berdasarkan hasil perhitungan *confusion matrix* pada tabel 4.14 penulis mendapatkan nilai akurasi, sensitivitas dan spesifitas adalah sebesar 99,77%, 99,63%, dan 99,90%.

#### 4.11. Hasil Analisis Penelitian Terkait

Berikut adalah perbandingan dengan penelitian yang berkaitan dengan klasifikasi NPDR menggunakan *transfer learning* CNN dan metode kombinasi CNN SVM ditunjukkan pada tabel 4.15 agar lebih memudahkan.

Tabel 4. 15. Perbandingan Penelitian Terkait

| Peneliti               | CNN Softmax (%) |        |           |           | CNN SVM (%) |        |           |           | Jumlah Kelas |
|------------------------|-----------------|--------|-----------|-----------|-------------|--------|-----------|-----------|--------------|
|                        | CNN Custom      | VGG 16 | Resnet 50 | Inception | CNN Custom  | VGG 16 | Resnet 50 | Inception |              |
| Qomariah, et al. 2019  | -               | -      | -         | -         | -           | 87,50  | 95,83     | 79,17     | 2            |
| Khojasteh, et al. 2019 | 90,60           | -      | -         | -         | -           | -      | 98,20     | -         | 2            |
| Wang, et al. 2018      | -               | 91,70  | -         | -         | -           | 80,6   | -         | -         | 4            |
| Araujo, et al., 2017   | 66,70           | -      | -         | -         | 65,00       | -      | -         | -         | 4            |
| Wang et al., 2018      | -               | 50,03  | -         | 63,23     | -           | -      | -         | -         | 5            |
| Wu et al., 2019        | -               | -      | 49,00     | 61,00     | -           | -      | -         | -         | 5            |
| purpose                | -               | 90,77  | 96,16     | 97,29     | -           | 98,42  | 99,77     | 99,09     | 4            |

Berbeda dengan penelitian yang dilakukan oleh Wang dan Araujo pada *multi class classification* menggunakan metode kombinasi CNN dan SVM memperoleh hasil klasifikasi CNN *softmax* mencapai akurasi yang lebih tinggi dibandingkan dengan metode kombinasi. Pada penelitian kami metode kombinasi CNN SVM memberikan akurasi melampaui metode klasifikasi CNN meskipun diterapkan pada klasifikasi multi kelas. Hal ini diperoleh karena kami menggunakan metode yang telah dioptimalkan pada penggunaan ekstraksi fitur. Yaitu dengan melakukan pelatihan ulang terhadap seluruh model pembelajaran transfer yang

digunakan (*pretrained*) untuk mendapatkan hasil akurasi yang tinggi dengan performa jaringan yang baik. Setelah itu model pelatihan yang dihasilkan disimpan yang akan di panggil kembali, untuk dilakukan klasifikasi dengan mengubah *fully layer CNN* digantikan dengan klasifikasi *mechine learning SVM*. Setelah itu kita buang bagian *fully connected layernya*. Fitur yang didapat dari ekstraksi fitur CNN nantinya dibentuk kedalam *dataframe* untuk selanjutnya diproses ke *classifier machine learning* lain dalam penelitian ini adalah SVM.

Hasil Penelitian kami menunjukkan hasil yang sama dengan penelitian yang telah dilakukan oleh Qomariah dengan menghasilkan nilai akurasi sebesar 95,24% untuk klasifikasi kombinasi CNN dan SVM pada model arsitektur Resnet50. Penelitian kami menghasilkan nilai akurasi sebesar 99,66% juga dengan menggunakan model arsitektur Resnet50. Namun penelitian kami mendapatkan nilai akurasi yang lebih tinggi, digunakan untuk *multi class classification* dengan menggunakan dataset yang lebih banyak. Dibandingkan dengan semua penelitian sebelumnya, nilai akurasi yang dihasilkan pada klasifikasi level keparahan NPDR pada penelitian ini menghasilkan akurasi yang lebih tinggi dengan menggunakan model *pretrained Resnet50* untuk ekstraksi fitur dan klasifikasi menggunakan SVM mencapai 99,77% untuk akurasi, 99,63% untuk sensitivitas, dan 99,90% untuk spesifitas.



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan serangkaian tahap uji coba, analisis dan pembahasan yang telah dilakukan dalam penelitian ini dapat disimpulkan bahwa:

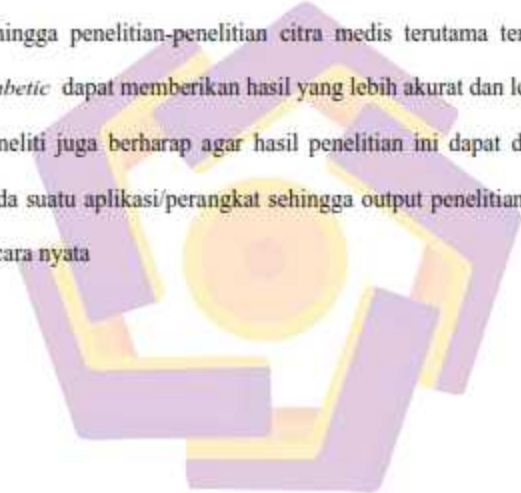
- a. Metode klasifikasi CNN SVM memberikan akurasi lebih baik dari pada metode klasifikasi dengan menggunakan CNN softmax.
- b. Metode CNN dapat memberikan nilai akurasi yang tinggi menggunakan ekstraksi fitur dengan model Resnet50 dengan nilai akurasi yaitu sebesar 99,77%
- c. Dari hasil perhitungan confusion matrix pada model CNN Resnet50 menghasilkan nilai akurasi, sensitivitas dan spesifitas adalah sebesar 99,77%, 99,85%, dan 99,93%

#### **5.2. Saran**

Ekstraksi fitur pada CNN jika dikombinasikan dengan klasifikasi menggunakan SVM dapat memberikan nilai akurasi yang lebih baik dibandingkan dengan metode klasifikasi dengan CNN. Untuk mendapatkan hasil pengujian yang benar-benar akurat beberapa saran yang penulis sampaikan antara lain :

1. Data dapat dipisahkan kedalam folder-folder pengelompokan data secara independent, pengelompokan data dapat dipisahkan per-folder data latih, data validasi dan data uji. Tujuannya adalah untuk benar-benar dapat dipastikan data uji tidak bocor pada saat pelatihan.

2. Penelitian lain juga dapat menggunakan metode preprocessing lainnya untuk mendapatkan citra yang baik agar proses ekstraksi fitur lebih dapat menonjolkan lesi atau jaringan abnormal, sehingga kesalahan prediksi pada proses klasifikasi menjadi lebih rendah.
3. Hasil penelitian ini juga belum diimplementasikan secara nyata. Di masa mendatang, peneliti berharap adanya dataset citra medis terutama citra fundus retina yang sudah tersedia dengan jumlah gambar yang lebih banyak sehingga penelitian-penelitian citra medis terutama tentang *retinopathy diabetic* dapat memberikan hasil yang lebih akurat dan lebih baik lagi.
4. Peneliti juga berharap agar hasil penelitian ini dapat diimplementasikan pada suatu aplikasi/perangkat sehingga output penelitian ini dapat terlihat secara nyata



## DAFTAR PUSTAKA

### PUSTAKA BUKU

- Cristianini N., Taylor J.S., 2000, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge Press University
- Davis, M.D., Blodi, B.A., 2006, Proliferative diabetic retinopathy. In: Ryan SJ, Schachat AP, Fourth Edi, St Louis, MO: Mosby
- Freeman, W.R., 1998, Practical Atlas of Retinal Disease and Therapy, Edition 2, page 199-213, Lippincott-Raven, Hongkong
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT press.
- Heaton, J. 2015. Artificial Intelligence for Humans : Deep learning and neural networks of Artificial Intelligence for Humans Series. Createspace Independent Publishing Platform
- Hinton, G., Sejnowski, T., 1999, Unsupervised Learning: Foundations of Neural Computation, MIT Press
- LeCun, Y., Bengio, Y., 1995, Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks. MIT Press
- Meyerle C.B., Chew E.Y., Ferris F.L., 2008, Nonproliferative Diabetic Retinopathy, In: Duh E.J. (eds) Diabetic Retinopathy, Contemporary Diabetes, Humana Press
- Russell, S. J., Norvig, P., 2010, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall
- Santosa, B. (2007). *Data mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Garah Ilmu.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., Riedmiller, M., 2015, Striving For Simplicity: The All Convolutional Net, ICLR
- Vapnik, V., Cortes, C., 1995. Support Vector Networks. Machine Learning, 20, 273-297
- Vaughan, D.G., Asbury, T., Eva, P.R., 2000, Oftalmologi Umum, Edisi 14, Widya Medika, Jakarta

## PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Ciulla, T. A., Amador, A. G., & Zinman, B., 2003, Diabetic Retinopathy and Diabetic Macular Edema: Pathophysiology, screening, and novel therapies. *Diabetes Care*, 26(9), 2653–2664
- Early Treatment Diabetic Retinopathy Studyresearch (ETDRS) group, 1985, Photocoagulation for diabetic macular edema. Early treatment diabetic retinopathy study report number 1, *Arch Ophthalmol*; 103:1796 – 1806.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning For Image Recognition. 1455 In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1456 pp. 770–778.
- Hubel, D.H., Wiesel, T.N., 1970, The period of susceptibility to the physiological effects of unilateral eye closure in kittens, *The Journal of Physiology*, 206 (2): 419–436, PMC 1348655, PMID 5498493
- Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012, Imagenet Classification With Deep Convolutional Neural Networks, *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Vol. 1, hal. 1097–1105.
- LeCun, Y., 1990, Handwritten Digit Recognition with a BackPropagation Network, *Advances in Neural Information Processing Systems 2*
- Samuel, A. L., 1959, Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210-229.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: *Proceedings1706 of the IEEE conference on computer vision and pattern recognition*. pp. 1–9
- Vedaldi, A., Lenc, K., 2015, MatConvNet: Convolutional Neural Networks for MATLAB. In *Proceedings of the 23 rd ACM International Conference on Multimedia* (hal. 689–692). New York, NY, USA: ACM.
- Wang, X., Lu, Y., Wang, Y., Chen, W., 2018, Diabetic Retinopathy Stage Classification Using Convolutional Neural Networks, *IEEE International Conference on Information Reuse and Integration (IRI)*, Salt Lake City, UTpp. 465-471.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017, Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700-4708.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M., 2019, Bag of Tricks for Image Classification with Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 558-567

## PUSTAKA LAPORAN PENELITIAN

Abramoff, M. D., Garvin, M. K., & Sonka, M. (2010). Retinal imaging and image analysis. *IEEE Reviews in Biomedical Engineering*, 3, 169–208.

Agustin, T., & Sunyoto, A. (2020). Optimization Convolutional Neural Network for Classification Diabetic Retinopathy Severity. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 66–71.

Agustin, T., Utami, E., & Fatta, H. Al. (2020). Implementation of Data Augmentation to Improve Performance CNN Method for Detecting Diabetic Retinopathy. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 83–88.

Akut, R. R. (2019). FILM: finding the location of microaneurysms on the retina. *Biomedical Engineering Letters*, 9(4), 497–506.

Albelwi, S., & Mahmood, A. (2017). A framework for designing the architectures of deep Convolutional Neural Networks. *Entropy*, 19(6).

Alex Krizhevsky, Ilya Sutskever, G. E. H. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*.

Alzubaidi, L., Fadhel, M. A., Al-shamma, O., Zhang, J., & Duan, Y. (2020). Deep learning models for classification of red blood cells in microscopy images to aid in sickle cell anemia diagnosis. *Electronics (Switzerland)*, 9(3).

Amel, F., Mohammed, M., & Abdelhafid, B. (2012). Improvement of the Hard Exudates Detection Method Used For Computer- Aided Diagnosis of Diabetic Retinopathy. *International Journal of Image, Graphics and Signal Processing*, 4(4), 19–27.

Araujo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., Polonia, A., & Campilho, A. (2017). Classification of breast cancer histology images using convolutional neural networks. *PLoS ONE*, 12(6), 1–14.

Bejiga, M. B., Zeggada, A., Nouffidj, A., & Melgani, F. (2017). A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery. *Remote Sensing*, 9(2).

Ben-Hur, A., & Weston, J. (2010). *A User's Guide to Support Vector Machines* (pp.

223–239).

- Biyani, R. S., & Patre, B. M. (2018). Algorithms for red lesion detection in Diabetic Retinopathy: A review. *Biomedicine and Pharmacotherapy*, 107(May), 681–688.
- Cardoso, J. S., Pinto Da Costa, J. F., & Cardoso, M. J. (2005). Modelling ordinal relations with SVMs: An application to objective aesthetic evaluation of breast cancer conservative treatment. *Neural Networks*, 18(5–6), 808–817.
- Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015). *Land Use Classification in Remote Sensing Images by Convolutional Neural Networks*. August.
- Christodoulidis, A., Hurtut, T., Tahar, H. Ben, & Cheriet, F. (2016). A multi-scale tensor voting approach for small retinal vessel segmentation in high resolution fundus images. *Computerized Medical Imaging and Graphics*, 52, 28–43.
- Chudzik, P., Majumdar, S., Calivá, F., Al-Diri, B., & Hunter, A. (2018). Microaneurysm detection using fully convolutional neural networks. *Computer Methods and Programs in Biomedicine*, 158, 185–192.
- Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.
- Decenciére, E., Zhang, X., Cazuguel, G., Laÿ, B., Cochener, B., Trone, C., Gain, P., Ordóñez-Varela, J. R., Massin, P., Erginay, A., Charton, B., & Klein, J. C. (2014). Feedback on a publicly distributed image database: The Messidor database. *Image Analysis and Stereology*, 33(3), 231–234.
- Eftekhari, N., Pourreza, H. R., Masoudi, M., Ghiasi-Shirazi, K., & Saeedi, E. (2019). Microaneurysm detection in fundus images using a two-step convolutional neural network. *BioMedical Engineering Online*, 18(1).
- F, F. T., Naresh, M., Ryan, P. F., & Cece, C. (2008). Accuracy of primary care clinicians in screening for diabetic retinopathy using single-image retinal photography. *Annals of Family Medicine*, 6(5), 428–434.
- Fleming, A. D., Goatman, K. A., Philip, S., Prescott, G. J., Sharp, P. F., & Olson, J. A. (2010). Automated grading for diabetic retinopathy: A large-scale audit using arbitration by clinical experts. *British Journal of Ophthalmology*, 94(12), 1606–1610.
- Fleming, A. D., Philip, S., Goatman, K. A., Olson, J. A., & Sharp, P. F. (2006). Automated microaneurysm detection using local contrast normalization and local vessel detection. *IEEE Transactions on Medical Imaging*, 25(9), 1223–1232.

- Fraz, M. M., Badar, M., Malik, A. W., & Barman, S. A. (2019). Computational Methods for Exudates Detection and Macular Edema Estimation in Retinal Images: A Survey. *Archives of Computational Methods in Engineering*, 26(4), 1193–1220.
- Fu, H., Xu, Y., Wong, D. W. K., & Liu, J. (2016). Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. *Proceedings - International Symposium on Biomedical Imaging, 2016-June*, 698–701.
- Gong, W., Chen, H., Zhang, Z., Zhang, M., Wang, R., Guan, C., & Wang, Q. (2019). A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion. *Sensors (Switzerland)*, 19(7).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT press.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- Indumathi, G., & Sathananthavathi, V. (2019). Microaneurysms detection for early diagnosis of diabetic retinopathy using shape and steerable gaussian features. In *Telemedicine Technologies: Big Data, Deep Learning, Robotics, Mobile and Remote Applications for Global Healthcare*. Elsevier Inc.
- Jebaseeli, T. J., Deva Durai, C. A., & Peter, J. D. (2019). Retinal blood vessel segmentation from diabetic retinopathy images using tandem PCNN model and deep learning based SVM. *Optik*, 199(October 2018), 163328.
- Katole, A. L., Yellapragada, K. P., Bedi, A. K., Kalra, S. S., & Siva Chaitanya, M. (2015). *Hierarchical Deep Learning Architecture for 10K Objects Classification*. December, 77–93.
- Khojasteh, P., Passos Júnior, L. A., Carvalho, T., Rezende, E., Aliahmad, B., Papa, J. P., & Kumar, D. K. (2019). Exudate detection in fundus images using deeply-learnable features. *Computers in Biology and Medicine*, 104, 62–69.
- Kim, J., Sangjun, O., Kim, Y., & Lee, M. (2016). Convolutional Neural Network with Biologically Inspired Retinal Structure. *Procedia Computer Science*, 88, 145–154.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2015). Alexnet\_Tugce\_Kyunghee. *Nips*.
- Lay, B. J., Baudoin, C. E., & Klein, J.-C. (1983). Automatic Recognition Of Microaneurysms In Diabetic Retinopathy. *Sixth International Congress for Stereology, 2/suppl 1*(November), 111–116.

- Lazar, I., & Hajdu, A. (2013). Retinal microaneurysm detection through local rotating cross-section profile analysis. In *IEEE Transactions on Medical Imaging* (Vol. 32, Issue 2, pp. 400–407).
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42(December 2012), 60–88.
- Long, S., Chen, J., Hu, A., Liu, H., Chen, Z., & Zheng, D. (2020). Microaneurysms Detection in Color Fundus Images based on Naive Bayesian Classification. *BioMedical Engineering OnLine*, 1–19.
- Maggiore, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 645–657.
- Manriquez, C. (2016). *Generalized Confusion Matrix for Multiple Classes* Generalized Confusion Matrix for Multiple Classes The total numbers of false negative (TFN), false positive (TFP), and true negative (TTN) for each class  $i$  will be calculated based on the Generalized. November, 5–7.
- Mazlan, N., Yazid, H., Arof, H., & Mohd Isa, H. (2020). Automated Microaneurysms Detection and Classification using Multilevel Thresholding and Multilayer Perceptron. *Journal of Medical and Biological Engineering*, 40(2), 292–306.
- Medhi, J. P., & Dandapat, S. (2016). An effective fovea detection and automatic assessment of diabetic maculopathy in color fundus images. *Computers in Biology and Medicine*, 74, 30–44.
- Melo, L. G. N., Morales, P. H., Drummond, K. R. G., Santos, D. C., Pizarro, M. H., Barros, B. S. V., Mattos, T. C. L., Pinheiro, A. A., Mallmann, F., Leal, F. S. L., Malerbi, F. K., & Gomes, M. B. (2018). Current epidemiology of diabetic retinopathy in patients with type 1 diabetes: A national multicenter study in Brazil. *BMC Public Health*, 18(1), 1–9.
- Mo, J., Zhang, L., & Feng, Y. (2018). Exudate-based diabetic macular edema recognition in retinal images using cascaded deep residual networks. *Neurocomputing*, 290, 161–171.
- Nørgaard, M. F., & Grauslund, J. (2018). Automated screening for diabetic retinopathy - A systematic review. *Ophthalmic Research*, 60(1), 9–17.
- Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabudhe,



- V., & Meriaudeau, F. (2018). Indian diabetic retinopathy image dataset (IDRiD): A database for diabetic retinopathy screening research. *Data*, 3(3), 1–8.
- Prentašić, P., & Lončarić, S. (2016). Detection of exudates in fundus photographs using deep neural networks and anatomical landmark detection fusion. *Computer Methods and Programs in Biomedicine*, 137, 281–292.
- Qomariah, D. U. N., Tjandrasa, H., & Faticah, C. (2019). Classification of diabetic retinopathy and normal retinal images using CNN and SVM. *Proceedings of 2019 International Conference on Information and Communication Technology and Systems, ICTS 2019*, 152–157.
- Rakshitha, T. R., Deepashree, D., & Prasanna, K. S. C. (2017). Comparative study of imaging transforms on diabetic retinopathy images. *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 - Proceedings*, 118–122.
- Rocha, A., Carvalho, T., Jelinek, H. F., Goldenstein, S., & Wainer, J. (2012). Points of interest and visual dictionaries for automatic retinal lesion detection. *IEEE Transactions on Biomedical Engineering*, 59(8), 2244–2253.
- Rosas-Romero, R., Martínez-Carballido, J., Hernández-Capistrán, J., & Uribe-Valencia, L. J. (2015). A method to assist in the diagnosis of early diabetic retinopathy: Image processing applied to detection of microaneurysms in fundus images. *Computerized Medical Imaging and Graphics*, 44, 41–53.
- Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1–2), 207–219.
- Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, 61, 85–117.
- Seoud, L., Hurtut, T., Chelbi, J., Cheriet, F., & Langlois, J. M. P. (2016). Red Lesion Detection Using Dynamic Shape Features for Diabetic Retinopathy Screening. *IEEE Transactions on Medical Imaging*, 35(4), 1116–1126.
- Shortliffe, E. H., & Blois, M. S. (2001). *The Computer Meets Medicine and Biology: Emergence of a Discipline*. 3–40.
- Siddiqi, R. (2019). Effectiveness of transfer learning and fine tuning in automated fruit image classification. *ACM International Conference Proceeding Series*, i, 91–100.
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

- Soomro, T. A., Gao, J., Khan, M. A. U., Khan, T. M., & Paul, M. (2016). Role of Image Contrast Enhancement Technique for Ophthalmologist as Diagnostic Tool for Diabetic Retinopathy. *2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016*.
- Srivastava, R., Duan, L., Wong, D. W. K., Liu, J., & Wong, T. Y. (2017). Detecting retinal microaneurysms and hemorrhages with robustness to the presence of blood vessels. *Computer Methods and Programs in Biomedicine*, 138(September 2017), 83–91.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June*, 1–9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 2818–2826.
- Ting, D. S. W., Cheung, G. C. M., & Wong, T. Y. (2016). Diabetic retinopathy: global prevalence, major risk factors, screening practices and public health challenges: a review. *Clinical and Experimental Ophthalmology*, 44(4), 260–277.
- Wan, S., Liang, Y., & Zhang, Y. (2018). Deep convolutional neural networks for diabetic retinopathy detection by image classification. *Computers and Electrical Engineering*, 72, 274–282.
- Wang, S., Tang, H. L., Al Turk, L. I., Hu, Y., Sanei, S., Saleh, G. M., & Peto, T. (2017). Localizing Microaneurysms in Fundus Images Through Singular Spectrum Analysis. *IEEE Transactions on Biomedical Engineering*, 64(5), 990–1002.
- Wang, X., Lu, Y., Wang, Y., & Chen, W. B. (2018). Diabetic retinopathy stage classification using convolutional neural networks. *Proceedings - 2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science, IRI 2018*, 465–471.
- Wang, Y., Sun, L., Ma, K., & Fang, J. (2018). Breast Cancer Microscope Image Classification Based on CNN with Image Deformation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10882 LNCS*. Springer International Publishing.
- Wu, B., Zhu, W., Shi, F., Zhu, S., & Chen, X. (2017). Automatic detection of microaneurysms in retinal fundus images. *Computerized Medical Imaging and*

*Graphics*, 55, 106–112.

- Zhang, L., Xu, L., Zhang, J. S., Zhang, Y. Q., Yang, H., & Jonas, J. B. (2012). Cotton-wool spot and optical coherence tomography of a retinal nerve fiber layer defect. *Archives of Ophthalmology*, 130(7), 913.
- Zhang, Y. D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., & Wang, S. H. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, 78(3), 3613–3632.
- Zhi, T., Duan, L. Y., Wang, Y., & Huang, T. (2016). Two-stage pooling of deep convolutional features for image retrieval. *Proceedings - International Conference on Image Processing, ICIP, 2016-Augus*, 2465–2469.
- Zhou, W., Wu, C., Chen, D., Yi, Y., & Du, W. (2017). Automatic Microaneurysm Detection Using the Sparse Principal Component Analysis-Based Unsupervised Classification Method. *IEEE Access*, 5, 2563–2572.

#### **PUSTAKA ELEKTRONIK**

- IDF, 21 April 2020, Diabetes Facts and Figures  
<https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html>
- Pickell, D., 14 Mei 2020, The Complete Guide to Machine Learning in 2020,  
<https://learn.g2.com/machine-learning>
- Stanford, 1 Mei 2020, Cs231n Convolutional Neural Network for visual recognition, <https://cs231n.github.io/convolutional-networks/#fc>