

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

##### 2.1.1 *Diabetic Retinopathy*

*Diabetic retinopathy (DR)* adalah komplikasi mikrovaskular umum dari diabetes mellitus dan merupakan penyebab utama kerugian visual pada lansia. Perkembangan retinopatik diabetes sangat berkorelasi dengan durasi diabetes yang lebih lama, hiperglikemia yang lebih besar, dan hipertensi [8].

Retinopatik diabetes disebabkan oleh tingginya kadar gula darah akibat diabetes. Seiring waktu, kelebihan gula dalam darah dapat merusak retina, bagian mata yang mendeteksi cahaya dan mengirim sinyal ke otak melalui saraf optik di bagian belakang mata. Diabetes merusak pembuluh darah di seluruh tubuh. Kerusakan pada mata dimulai ketika gula dalam darah menyebabkan perubahan pada pembuluh darah kecil yang menuju retina. Perubahan ini membuat aliran darah menjadi lebih sulit, yang mengakibatkan pembuluh darah tersumbat, bocor, atau berdarah. Untuk mengatasi pembuluh darah yang tersumbat, mata kemudian menumbuhkan pembuluh darah baru yang tidak berfungsi dengan baik. Pembuluh darah baru ini bisa mudah bocor atau berdarah [13].

##### 2.1.2 *Fundus Image*

*Fundus image* (citra fundus) merupakan dokumentasi visual yang mendokumentasikan tampilan retina pasien yang diperiksa melalui oftalmoskopi. *Fundus image* digunakan dalam mendokumentasikan DR karena detail retina lebih mudah untuk divisualisasikan dibandingkan dengan pemeriksaan langsung [14]. Pada penelitian ini, *fundus image* akan digunakan untuk pelatihan dan pengujian pada model yang dikembangkan. Gambar yang digunakan diperoleh melalui laman kaggle [15] yang berisi lima tingkatan DR

diantaranya yaitu *normal*, *mild*, *Moderate*, *severe* dan *proliferative diabetic retinopathy* (PDR).

### **2.1.3 Deep learning**

*Deep learning* merupakan bagian dari machine learning yang menggunakan jaringan saraf multi-lapisan (*deep neural networks*) untuk meniru kemampuan pengambilan keputusan kompleks dari otak manusia. Pembelajaran mendalam ini dicirikan oleh beberapa lapisan pemrosesan yang disusun untuk mengekstraksi fitur yang semakin rumit dari data. Arsitektur hierarkis dari jaringan pembelajaran mendalam memungkinkan mereka untuk belajar dan merepresentasikan pola dan fitur yang kompleks secara mandiri saat mereka melalui lapisan-lapisan tersebut. [16], [17]. *Deep learning* memungkinkan pengembangan model komputasi dengan lapisan pemrosesan yang lebih dalam dan telah menghasilkan kemajuan yang signifikan dalam bidang pengenalan objek visual. *Deep learning* menemukan pola kompleks dalam kumpulan data besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin dapat mengubah parameter internalnya dalam menghasilkan representasi pada setiap lapisan berdasarkan representasi pada lapisan sebelumnya [18].

### **2.1.4 Transfer Learning**

Inovasi terbaru dalam *deep learning* adalah model *transfer learning*. Metode ini memanfaatkan jaringan neural yang telah dilatih sebelumnya dan mengurangi jumlah parameter dengan mengambil beberapa bagian dari model yang sudah ada untuk diterapkan pada model baru. *Transfer learning* memungkinkan pemanfaatan model *deep learning* untuk menyelesaikan masalah baru dengan lebih efisien [9].

### **2.1.5 EfficientNet**

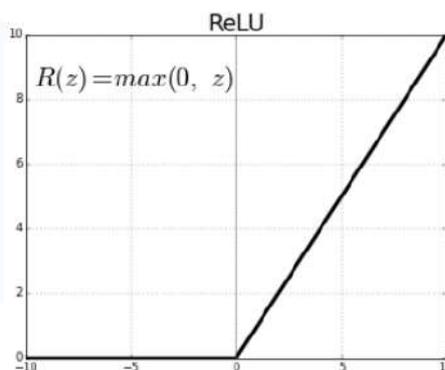
EfficientNet adalah jenis arsitektur jaringan saraf konvolusional yang membedakan dirinya dengan metode penskalanya yang terstruktur. Metode ini secara seragam meningkatkan semua dimensi jaringan, termasuk kedalaman,

lebar, dan resolusi, menggunakan sebuah koefisien gabungan. Penskalaan ini didasarkan pada keyakinan bahwa gambar yang lebih besar membutuhkan jaringan dengan lapisan yang lebih dalam dan lebih banyak saluran untuk menangkap pola yang lebih kompleks. Base model EfficientNet-B0 sendiri terdiri dari blok residual botol terbalik dari MobileNetV2 dan blok *squeeze-and-excitation*. Keunggulan lainnya adalah kemampuannya dalam mentransfer pengetahuan dengan baik, menghasilkan akurasi yang tinggi dengan jumlah parameter yang lebih sedikit pada berbagai dataset *transfer learning* seperti CIFAR-100 dan Flowers [19].

### 2.1.6 Rectified Linear Unit (ReLU)

*Rectified Linear Unit* (ReLU) adalah jenis fungsi aktivasi yang sering digunakan dalam model *convolutional neural network* ataupun *deep learning*. Fungsi ini menghasilkan nilai 0 jika *input*-nya negatif, dan jika *input*-nya positif, maka fungsi akan mengembalikan nilai *input* tersebut [20]. Fungsi ReLU dapat dijelaskan dengan rumus berikut dan bisa diilustrasikan pada Gambar 2.1.

$$f(x) = \max(0, x)$$



Gambar 2.1 Diagram Fungsi ReLU

Sumber: [21]

Keunggulan Fungsi ReLU membuatnya menjadi pilihan umum dalam berbagai bidang. Keunggulannya terletak pada kesederhanaan fungsinya yang menghasilkan komputasi yang cepat. Selain itu, ReLU mampu mengatasi

masalah umum yang sering terjadi pada fungsi aktivasi lain, seperti masalah gradien yang hilang, sehingga membantu dalam pelatihan model *deep learning* tanpa mengurangi performa. Kelebihan lainnya adalah ReLU tidak memiliki batasan atas, sehingga dapat mempercepat proses konvergensi selama pelatihan jaringan [22].

### **2.1.7 Adam**

Optimisasi Adam adalah metode stochastic gradient descent yang didasarkan pada estimasi adaptif dari momen orde pertama dan orde kedua. Menurut Kingma et al., [23], metode ini "efisien dalam penggunaan komputasi, membutuhkan sedikit memori, tetap stabil terhadap penskalaan ulang diagonal gradien, dan sangat sesuai untuk masalah yang melibatkan data atau parameter dalam jumlah besar". *Learning rate* yang digunakan pada penelitian ini yaitu sebesar 0,0001 untuk menambah akurasi tanpa menyebabkan pelatihan model memakan waktu yang lama.

### **2.1.8 Keras Framework**

Keras merupakan API tingkat tinggi dari Tensorflow pada bahasa pemrograman Python yang menyediakan antarmuka ramah pengguna dengan fokus pada *deep learning* modern [24]. Keras adalah sebuah perangkat lunak sumber terbuka dalam bahasa pemrograman Python yang telah disahkan dengan lisensi dari *Massachusetts Institute of Technology* (MIT). Banyak pihak telah berkontribusi dalam pengembangannya, termasuk Google, Microsoft, dan Francois Chollet, yang saat ini menjadi pemegang lisensi resmi untuk paket Keras itu sendiri. Paket ini menyediakan berbagai fitur yang meliputi beberapa model klasifikasi gambar yang telah dilatih sebelumnya dan sering digunakan [25].

### **2.1.9 Tensorflow**

TensorFlow adalah kerangka kerja sumber terbuka untuk *deep learning* dan aliran data, yang memiliki antarmuka pemrograman aplikasi (API) untuk analisis suara, pemrosesan bahasa alami, dan visi komputer [26]. Khususnya,

API deteksi objek TensorFlow di bidang visi komputer telah banyak diterapkan pada teknologi di bidang pertanian, teknik, dan kedokteran. Namun, hambatan untuk menggunakan kerangka kerja ini masih tinggi melalui antarmuka baris perintah (CLI) dan kode, terutama bagi pemula dan amatir di bidang teknologi informasi (IT).

#### **2.1.10 CLAHE**

*Contrast Limited Adaptive Histogram Equalization* (CLAHE) adalah sebuah teknik yang memperbaiki masalah kontras rendah pada gambar digital, terutama gambar medis. Dalam penggunaannya pada bidang citra medis, CLAHE menunjukkan hasil yang lebih baik daripada *Adaptive Histogram Equalization* (AHE) dan *Histogram Equalization* (HE) biasa. CLAHE beroperasi dengan membatasi peningkatan kontras yang biasanya dihasilkan oleh HE biasa, yang dapat menyebabkan peningkatan noise. Dengan membatasi peningkatan kontras ini, CLAHE berhasil memberikan hasil yang diinginkan pada kasus di mana noise menjadi terlalu dominan akibat peningkatan kontras, terutama pada gambar medis. Secara umum, peningkatan kontras dapat dijelaskan sebagai kecuraman fungsi yang menghubungkan nilai intensitas gambar *input* dengan intensitas gambar hasil yang diinginkan. Kontras dapat dikontrol dengan membatasi kecuraman fungsi tersebut. Selain itu, peningkatan kontras secara langsung berkaitan dengan tinggi histogram pada nilai intensitas tertentu [27]. Pada penelitian ini, penerapan CLAHE dilakukan menggunakan *library* OpenCV dengan fungsi `createCLAHE`.

#### **2.1.11 Enhanced Green**

*Preprocessing enhanced green* menggunakan dua teknik pengolahan citra, yaitu CLAHE dan unsharp masking, pada citra dari kanal hijau. CLAHE meningkatkan kontras dengan membagi citra menjadi beberapa wilayah kontekstual berukuran  $8 \times 8$ , lalu menerapkan histogram equalization pada setiap wilayah tersebut [28]. Pendekatan ini memungkinkan penyesuaian kontras yang adaptif di berbagai bagian citra, menghasilkan peningkatan kontras yang lebih baik daripada metode equalization histogram global.

Selanjutnya, teknik *unsharp masking* digunakan untuk meningkatkan ketajaman gambar dengan cara menghasilkan versi yang "terasa" dari gambar asli. Ini dilakukan dengan membuat salinan gambar yang diperhalus (atau "blurred") menggunakan filter Gaussian, lalu mengurangkan gambar yang diperhalus ini dari gambar asli. Hasilnya adalah gambar yang lebih tajam dengan detail yang lebih jelas.

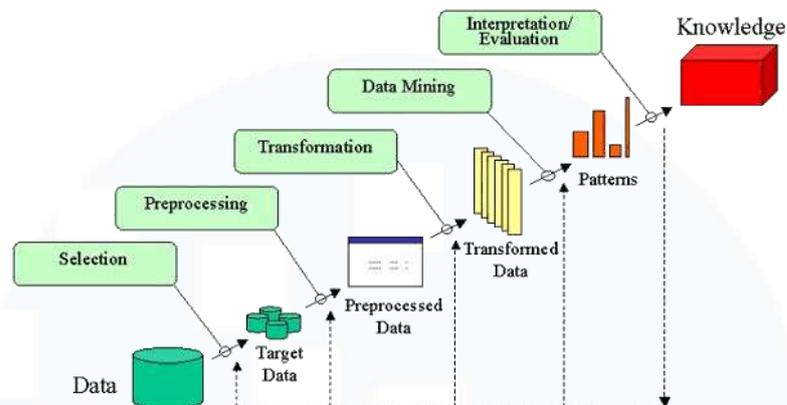
Dengan mengombinasikan kedua teknik ini dalam *preprocessing*, citra *Enhanced Green* dapat diolah dengan lebih baik, meningkatkan kualitas visual dan kegunaannya dalam aplikasi berbasis citra. *Library* yang digunakan yaitu OpenCV dengan fungsi *GaussianBlur* dan *addWeighted*

#### **2.1.12 ROC Curve**

*Receiver Operating Characteristic (ROC) curve* adalah metode analisis yang direpresentasikan dalam bentuk grafik yang digunakan untuk mengevaluasi kinerja suatu metode klasifikasi diagnostik biner [29]. Kurva AUC-ROC adalah alat pengukuran kinerja yang digunakan dalam masalah klasifikasi untuk mengevaluasi seberapa baik model dapat membedakan antara kelas-kelas yang berbeda. Nilai AUC (*Area Under the Curve*) menggambarkan seberapa baik model dapat membedakan antara pasien yang memiliki penyakit dan yang tidak memiliki penyakit. Semakin tinggi nilai AUC, semakin baik model dalam memprediksi dengan benar. Kurva ROC digambar dengan TPR terhadap FPR di mana TPR berada pada sumbu y dan FPR berada pada sumbu x [30].

#### **2.1.13 KDD**

*Knowledge discovery in databases (KDD)* merupakan proses multi-tahap yang bersifat iteratif untuk mengekstraksi, mengubah, dan membersihkan data dan pola dari database mentah menjadi pengetahuan yang berharga [31]. Metode KDD berisi beberapa proses *data selection*, *pre-processing*, *transformation*, *data mining*, dan *interpretation/evaluation*.



Gambar 2.2 Tahapan dalam proses KDD

Sumber: [32]

Tahapan selection mencakup proses pemilihan dataset target atau pemilihan subset variabel dan sampel data yang ingin dieksplorasi untuk menemukan pengetahuan baru. Setelah dataset target ditentukan, proses berlanjut ke tahap pra-pemrosesan, yang meliputi pembersihan data dan penyesuaian lainnya untuk memastikan data lebih konsisten. Setelah data menjadi konsisten, tahap transformasi dilakukan. Dalam tahap ini, data diubah menggunakan berbagai metode, seperti pengurangan dimensi, untuk memudahkan pemrosesan pada tahap selanjutnya. Tahap berikutnya adalah penambangan data, yang melibatkan pencarian pola dalam data yang sesuai dengan tujuan awal yang telah ditetapkan. Terakhir, ada tahap evaluasi, di mana pola-pola yang ditemukan diinterpretasikan untuk memastikan informasi yang berguna telah diekstraksi [32].

#### 2.1.14 Python

Python adalah bahasa pemrograman tingkat tinggi yang sangat fleksibel dan mudah dipelajari. Dengan struktur data bawaan yang kuat dan tipenya yang dinamis, serta sintaksis yang sederhana, Python mendukung pengembangan aplikasi cepat dan pemeliharaan kode yang efisien dengan menyediakan modul dan paket sehingga mendorong modularitas program dan penggunaan kembali kode [33]. Pada penelitian ini, Python digunakan karena telah memiliki *library* yang tersedia untuk digunakan pada pembangunan model.

### 2.1.15 Confusion Matrix

*Confusion matrix* merupakan sebuah alat evaluasi kinerja dalam pembelajaran mesin yang merepresentasikan akurasi dari model klasifikasi. Dalam matrix ini, terdapat empat elemen utama yang menggambarkan hasil klasifikasi model. *True Positives* (TP) adalah ketika model benar-benar mengenali kasus sebagai positif ketika memang benar-benar positif. Sebaliknya, *True Negatives* (TN) menunjukkan ketika model benar-benar mengenali kasus sebagai negatif ketika memang benar-benar negatif. *False Positives* (FP) adalah ketika model salah mengklasifikasikan kasus sebagai positif ketika seharusnya negatif, juga dikenal sebagai kesalahan Tipe I. Sedangkan *False Negatives* (FN) terjadi ketika model salah mengklasifikasikan kasus sebagai negatif ketika seharusnya positif, dikenal sebagai kesalahan Tipe II. Informasi dari *confusion matrix* dapat digunakan untuk menghitung berbagai metrik evaluasi kinerja model seperti akurasi, presisi, *recall*, dan *f1-score* [34].

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2.3 *Confusion matrix*

Sumber: [35]



Gambar 2.4 Hasil sebenarnya dan prediksi

Sumber: [36]

Akurasi adalah ukuran umum untuk mengevaluasi kinerja model klasifikasi dan dihitung sebagai persentase dari prediksi yang benar (baik positif maupun negatif) dari total data yang dievaluasi. Berikut merupakan rumus yang digunakan dalam menghitung akurasi.

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

Gambar 2.5 Rumus Akurasi

Presisi mengukur seberapa akurat model dalam mengidentifikasi instance positif. Dalam konteks klasifikasi, presisi mengukur berapa proporsi dari kasus yang diprediksi sebagai positif oleh model yang benar-benar positif. Dengan kata lain, presisi memberi tahu kita seberapa sedikit kasus negatif yang salah diklasifikasikan sebagai positif oleh model. Gambar 2.6 merupakan rumus yang digunakan untuk menghitung presisi. Di mana TP adalah jumlah True Positives (kasus positif yang benar diprediksi oleh model) dan FP adalah jumlah False Positives (kasus negatif yang salah diprediksi sebagai positif oleh model).

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

Gambar 2.6 Rumus presisi

*Recall* mengukur seberapa baik model dapat menemukan kembali semua kasus positif. Dalam konteks klasifikasi, recall mengukur berapa proporsi dari total kasus positif yang benar-benar ditemukan oleh model.

Dengan kata lain, *recall* memberi tahu kita seberapa sedikit kasus positif yang terlewatkan oleh model. Gambar 2.7 merupakan rumus *recall* di mana TP adalah jumlah *True Positives* (kasus positif yang benar diprediksi oleh model) dan FN adalah jumlah *False Negatives* (kasus positif yang salah diprediksi sebagai negatif oleh model).

$$\text{Recall} = \frac{\text{True Positive}(TP)}{\text{True Positive}(TP) + \text{False Negative}(FN)}$$

Gambar 2.7 Rumus *recall*

*F1-score* adalah sebuah metrik yang digunakan dalam evaluasi kinerja model klasifikasi. Ini adalah rata-rata harmonis dari presisi (*precision*) dan *recall*, memberikan ukuran yang seimbang antara kedua metrik tersebut. *F1-score* berguna ketika ingin mencapai keseimbangan antara presisi dan *recall*, terutama saat distribusi kelas tidak seimbang. Tujuan dari *F1-score* adalah untuk memberikan gambaran yang lebih holistik tentang kinerja model klasifikasi daripada hanya menggunakan presisi atau *recall* secara terpisah. Dengan memperhitungkan kedua aspek tersebut, *F1-score* memberikan indikasi tentang seberapa baik model mampu mengidentifikasi kasus positif dan seberapa baik model mampu menemukan kembali semua kasus positif dalam data. Berikut merupakan rumus yang digunakan untuk menghitung *f1-score*.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Gambar 2.8 Rumus *f1-score*

## 2.2 Penelitian Terdahulu

Salah satu tahapan penelitian ini adalah tahap pra-KDD yang meliputi kegiatan penelitian dan analisis penelitian-penelitian sebelumnya, pengumpulan informasi terkait topik, peralatan dan metode penelitian serta pemilihan metode penelitian.

Tabel 2.1 berikut merupakan penelitian terdahulu yang dijadikan sebagai pembandingan pada penelitian ini.

Tabel 2.1 Tabel Penelitian Terdahulu

<b><i>Development of revised ResNet-50 for diabetic retinopathy detection</i></b> [11]	
Penulis	Chun-Ling Lin, Kun-Chi Wu
Tahun	2023
Jurnal	BMC Bioinformatics
Model	<i>Revised ResNet-50</i>
Dataset	Kaggle DR dataset
Hasil	Model ResNet-50 yang diajukan memberikan akurasi pelatihan senilai 83,95% dan akurasi uji senilai 74,32% yang lebih baik daripada model CNN lainnya.
Saran	Diperlukan analisis lebih lanjut pada model CNN lainnya menggunakan <i>transfer learning</i> dan <i>hyperparameter-tuning</i> untuk meningkatkan performa klasifikasi citra fundus.
<b><i>Using transfer learning for diabetic retinopathy stage classification</i></b> [12]	
Penulis	Enas M.F. El Houby
Tahun	2021
Jurnal	<i>Applied Computing and Informatics</i>
Model	VGG-16
Dataset	<i>Kaggle retinal fundus image dataset (DDR)</i>
Hasil	Akurasi pada klasifikasi dua kelas yaitu 86,5%. Pada klasifikasi tiga kelas yaitu 80,5%. Pada klasifikasi empat kelas mendapatkan 63,5%. Pada klasifikasi lima kelas yaitu 73,7%.
Saran	Diperlukan penelitian untuk mendapatkan teknik atau model yang dapat memberikan hasil lebih akurat dalam membedakan antara kelas-kelas yang berbeda.
<b><i>Application of Deep Convolutional Neural Networks VGG-16 and GoogLeNet for Level Diabetic Retinopathy Detection</i></b> [37]	
Penulis	Chaicana Suedumrong, Komgrit Leksakul, Pranprach Wattana, Poti Chaopaisarn
Tahun	2022
Jurnal	<i>Lecture Notes in Networks and Systems</i>
Model	VGG-16 and GoogLeNet
Dataset	EyePACS
Hasil	Model <i>transfer learning</i> VGG-16 memberikan tingkat akurasi yang lebih tinggi (71,65%) dibandingkan dengan model GoogLeNet (61,58%).
Saran	Menggunakan teknik baru pada model untuk meningkatkan akurasi serta mempertimbangkan waktu yang dibutuhkan pada pelatihan model
<b><i>Diabetic Retinopathy Detection Using Transfer Learning and Deep Learning</i></b> [38]	
Penulis	Akhilesh Kumar Gangwar, Vadlamani Ravi
Tahun	2021
Jurnal	Advances in Intelligent Systems and Computing
Algoritma	Inception-ResNet-v2
Dataset	Messidor-1 and APTOS dataset
Hasil	Mendapatkan akurasi uji sebesar 72,33% pada Messidor-1 dan 82,18% pada APTOS.
Saran	Menggunakan berbagai model transfer learning untuk meningkatkan kinerja model.

Pengaruh Preprocessing Terhadap Klasifikasi Diabetic Retinopathy dengan Pendekatan Transfer Learning Convolutional Neural Network [28]	
Penulis	Juan Elisha Widyaya, Setia Budi
Tahun	2021
Jurnal	Jurnal Teknik Informatika dan Sistem Informasi
Model	Inception v3
Dataset	APTOS 2019 Blindness Detection
Hasil	<i>Preprocessing</i> menggunakan CLAHE memberikan akurasi tertinggi sebesar 78,89%, presisi senilai 82,97%, <i>recall</i> senilai 74,64%, dan AUC sebesar 95,81%.
Saran	Melakukan <i>hyperparameter tuning</i> dan menggunakan beberapa teknik optimasi pelatihan model.

Penelitian [11] bertujuan untuk menunjukkan dampak dari penerapan SOP pada pengolahan data DR, dengan menggunakan model ResNet-50 pada dataset EyePACS. Penelitian ini memberikan prosedur standar yang dapat dilakukan pada tahap *preprocessing* citra fundus. Tahapan tersebut dimulai dengan menghapus area hitam pada citra fundus. Kemudian, akan dilakukan pemotongan berbentuk lingkaran pada gambar. Setelah itu akan dilakukan pengecekan kualitas pada *fundus image*. Gambar dengan kualitas yang sesuai akan diolah kembali menggunakan *histogram equalization* untuk memperjelas gambar. Model Revised ResNet-50 yang diajukan pada penelitian ini memberikan akurasi uji senilai 74%. Saran yang diberikan pada penelitian ini yaitu melakukan analisa lebih lanjut pada model CNN yang berbeda menggunakan *transfer learning* dan *hyperparameter-tuning* untuk meningkatkan performa klasifikasi.

Penelitian [12] menggunakan pendekatan *transfer learning* pada model VGG-16 untuk melakukan deteksi DR. Percobaan dilakukan berulang kali dengan mengubah jumlah kategori dari tingkatan DR. Berdasarkan hasil evaluasi, model yang dibuat tanpa melakukan augmentasi mampu memberikan hasil akurasi yang lebih baik. Akan tetapi, model tersebut memberikan hasil yang *overfitted*. Selain VGG-16, penelitian ini juga menguji model lain yaitu ResNet50 dan Inception. Ketiga model tersebut memberikan hasil akurasi yang sama senilai 75% pada klasifikasi DR lima kelas. Berdasarkan pengujian, ditemukan bahwa batas antara kelas-kelas yang berbeda tidak begitu jelas, terutama antara yang *mild* dan normal, serta antara kelas *severe* dan

*proliferative*. Penelitian ini menyarankan untuk mencari teknik atau model yang lebih akurat dalam membedakan tingkatan DR.

Penelitian [37] menggunakan arsitektur CNN dan metode *transfer learning* menggunakan model GoogLeNet dan VGG-16. Model ini dilatih menggunakan data EyePACS [39] yang berjumlah 88,702 gambar. Hasil dari eksperimen saat menggunakan model GoogLeNet dengan fungsi Sigmoid sebagai fungsi aktivasi dan *learning rate* senilai 0,01 memberikan akurasi sebesar 61,58%. Adapun model VGGNet-16 dengan *fine tuning*, memberikan akurasi sebesar 71,65%. Penelitian ini menyarankan untuk menggunakan teknik baru untuk meningkatkan akurasi dengan mempertimbangkan waktu pelatihan. Eksperimen juga dapat dilakukan pada gambar berwarna biru atau merah sebagai *input* ke dalam model.

*Transfer learning* juga digunakan pada penelitian [38]. *Transfer learning* digunakan pada model Inception-ResNet-v2 yang telah dilatih, kemudian *custom block* CNN ditambahkan untuk membangun model hybrid. Model tersebut kemudian dievaluasi pada dataset Messidor-1 dan APTOS 2019. Model yang dibuat memberikan akurasi 6,3% lebih besar dibandingkan dengan model GoogLeNet. Pada dataset Messidor-1, model memberikan akurasi sebesar 72,33%. Sedangkan pada dataset APTOS, model memberikan akurasi sebesar 82,18%. Saran dari penelitian ini yaitu melakukan penyeimbangan pada distribusi data dan melakukan penelitian lebih lanjut pada *transfer learning* untuk meningkatkan performa secara keseluruhan.

Pada penelitian [28], tahapan *preprocessing* dijadikan fokus penelitian untuk melihat dampaknya terhadap akurasi model. Dari keempat metode *preprocessing* yang telah diuji, termasuk Graham, Nakhon, Ramasubramanian, dan *enhanced green*, ditemukan bahwa metode *enhanced green* menghasilkan kinerja terbaik dengan tingkat akurasi mencapai 76,10% dan 78,79% setelah *fine tuning*. Penelitian berikutnya dapat menerapkan *hyperparameter tuning* serta penerapan berbagai teknik untuk mengoptimalkan pelatihan model.