

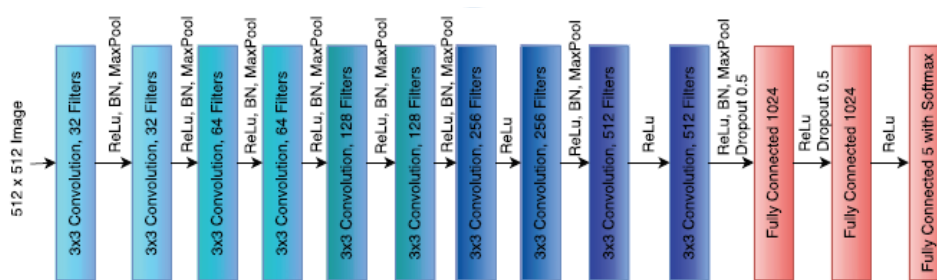
## BAB 3 TINJAUAN PUSTAKA

Berbagai penelitian sudah dilakukan untuk membangun model menggunakan komputer untuk mendiagnosis DR secara otomatis [2]. Metode tradisional *machine learning* akan mengekstrak fitur dari citra *fundus* retina. Fitur yang sudah diekstrak tersebut menjadi input bagi *classifier* tertentu seperti *random forest*, *support vector machine*, atau *AdaBoost classifier* [30]. Salah satu prasyarat teknik *hand engineered feature* adalah memahami gejala klinis DR secara mendalam untuk memilih fitur yang tepat. Berbeda dengan *deep learning* seperti CNN yang akan mengekstrak fitur secara otomatis dari citra [16]. Jika dibandingkan dengan *hand engineered feature*, teknik *deep learning* secara umum memberikan hasil yang lebih unggul [2]. Berbagai teknik dan algoritma *deep learning* sudah diterapkan untuk mengklasifikasikan citra *fundus* retina. Pada bagian berikutnya mengulas beberapa penelitian yang sudah dilakukan terkait klasifikasi DR dengan teknik *deep learning* CNN. Hal-hal yang ditinjau adalah dataset yang digunakan, tugas klasifikasi yang dilakukan, teknik *preprocessing* citra yang dipakai, arsitektur yang digunakan, teknik *training* yang dilakukan, *hyperparameter* untuk *training* model, dan hasil penelitian yang bersangkutan.

### 3.1 Penelitian Klasifikasi DR dengan *Deep Learning* CNN

Harry Pratt dan kawan-kawan [31] menggunakan dataset EyePACS untuk mengklasifikasikan DR ke dalam lima kelas. *Preprocessing* yang dilakukan adalah mengubah ukuran citra menjadi  $512 \times 512$  *pixel* dan normalisasi warna. Arsitektur CNN yang digunakan terdiri dari 10 set lapisan *convolutional*, aktivasi *leaky ReLU*, *batch normalization*, dan *max pooling*. Pada bagian akhir dari arsitektur CNN diikuti dengan 3 lapisan *fully connected* seperti yang ditunjukkan oleh Gambar 3.1. Harry Pratt menggunakan regularisasi *dropout* dan mengimplementasikan *class weight* yang merupakan perbandingan jumlah citra dari masing-masing kelas untuk mengurangi *overfitting*. Bobot model diinisialisasi dengan inisialisasi Gaussian dan model di-*training* dengan pendekatan *end to end learning*. Parameter yang dipakai untuk *training* model adalah: optimasi *Stochastic Gradient Descent Momentum* (SGDM) dengan Nesterov, *learning rate* 0.0001 untuk

5 *epoch* pertama, lalu *learning rate* dinaikan menjadi 0.0003 untuk 120 *epoch* berikutnya. Setiap kali model saturasi (nilai *loss* dan akurasi tidak berubah) *learning rate* diturunkan dengan faktor 10. Teknik augmentasi citra juga dilakukan selama proses *training* yaitu rotasi citra secara acak antara  $0^\circ - 90^\circ$ , pembalikan citra horizontal dan vertikal secara acak, dan penggeseran horizontal maupun vertikal secara acak. Hasil yang didapat adalah 75% *accuracy*, 95% *specificity*, 30% *sensitivity*.



Gambar 3.1Arsitektur Harry Pratt [31]

Darshit Doshi dan kawan-kawan [32] menggunakan dataset dari EyePACS untuk mengklasifikasikan DR ke dalam lima kelas. Mereka melakukan *preprocessing* dengan mengambil *channel* hijau dari citra RGB, mengubah ukuran citra menjadi  $512 \times 512$  *pixel*, meningkatkan kontras citra agar merata pada setiap *pixel* dengan *histogram equalization*, dan normalisasi citra dengan *min-max*. Arsitektur CNN yang digunakan terdiri dari lima set kombinasi lapisan *convolution*, *max pooling*, dan *dropout*. Pada bagian akhir dari arsitektur CNN diikuti dengan dua set lapisan *fully connected* dan *pooling*. Arsitektur ini bisa dilihat pada Tabel 3.1. Fungsi aktivasi yang mereka gunakan adalah *leaky ReLU*. Model di-*training* dengan pendekatan *end to end learning* dan bobot model diinisialisasi dengan menggunakan inisialisasi Glorot yang disample dari distribusi seragam. Model di-*training* sebanyak 250 *epoch* menggunakan optimasi SGDM (momentum = 0.9) dengan Nesterov. *Learning rate* untuk 200 *epoch* pertama adalah 0.003 sedangkan 50 *epoch* sisanya menggunakan *learning rate* 0.0003. Selama *training* dilakukan augmentasi citra dengan transformasi seperti *shear*, *flop*, *transverse*, dan *transpose*. Hasil percobaan Darshit Doshi bisa dilihat pada Tabel 3.2.

**Tabel 3.1 Arsitektur CNN Darshit Doshi [32]**

<b>Layers</b>	<b>Model 1</b>	<b>Model 2</b>	<b>Model 3</b>
<i>input</i>	1x512x512	1x512x512	1x512x512
conv 1	16x256x256	16x256x256	16x256x256
conv 2	16x256x256	16x256x256	16x256x256
pool 1	16x128x128	16x128x128	16x128x128
dropout 1	16x128x128	16x128x128	16x128x128
conv 3	32x64x64	32x64x64	32x64x64
conv 4	32x64x64	32x64x64	32x64x64
pool 2	32x32x32	32x32x32	32x32x32
dropout 2	32x32x32	32x32x32	32x32x32
conv 5	48x32x32	64x32x32	64x32x32
conv 6	48x32x32	64x32x32	64x32x32
conv 7	48x32x32	64x32x32	64x32x32
pool 3	48x16x16	64x16x16	64x16x16
dropout 3	48x16x16	64x16x16	64x16x16
conv 8	64x16x16	128x16x16	96x16x16
conv 9	64x16x16	128x16x16	96x16x16
conv 10	64x16x16	128x16x16	96x16x16
pool 4	64x8x8	128x8x8	96x8x8
dropout 4	64x8x8	128x8x8	96x8x8
conv 11	128x8x8	256x8x8	128x8x8
conv 12	128x8x8	256x8x8	128x8x8
pool 5	128x4x4	256x4x4	128x4x4
dropout 5	128x4x4	256x4x4	128x4x4
hidden 1	400	256	256
maxout 1	200	128	128
dropout 6	200	128	128
hidden 2	400	256	256
maxout 2	200	128	128
<i>output</i>	5	5	5

**Tabel 3.2 Hasil Percobaan Darshit Doshi [32]**

<b>Model</b>	<b>Quadratic kappa score</b>
Model 1	0.3066
Model 2	0.35
Model 3	0.386

Maria A. Bravo dan Pablo A. Arbelaez [8] menggunakan dataset dari EyePacs untuk mengklasifikasikan DR ke dalam beberapa kategori klasifikasi, mulai dari biner, tiga kelas, dan lima kelas. Bagian yang disoroti pada tinjauan kali ini adalah klasifikasi lima kelas dengan mengesampingkan klasifikasi biner maupun tiga kelas. *Preprocessing* yang dilakukan adalah:

1. Circle RGB: citra *fundus* retina RGB yang dipotong melingkar dengan ukuran  $224 \times 224 \text{ pixel}$ .
2. Square: citra *fundus* retina yang dipotong bujur sangka dengan ukuran  $224 \times 224 \text{ pixel}$ .
3. Color centered: citra pada nomor 1 dikurangi dengan warna rata-rata lokal menggunakan filter Gaussian.
4. Gray scale: citra nomor 1 dikonversi menjadi citra *gray scale*.

Arsitektur CNN yang digunakan adalah VGG-16 dan Inception V4. Kedua arsitektur ini di-*training* dengan pendekatan *transfer learning*. Model VGG-16 di-*training* kira-kira sebanyak 14 *epoch*, *learning rate* 0.001 yang diubah setiap 5 *epoch* dengan faktor 10, *batch size* 115, optimasi SGDM (momentum = 0.9), dan regularisasi *dropout* untuk mengurangi *overfitting*. Sedangkan model Inception v4 di-*training* sebanyak 20 *epoch*, *batch size* 32, *learning rate* 0.001, dan regularisasi *weight decay* 0.00004. Selama *training* model, baik arsitektur VGG-16 maupun arsitektur Inception v4, dilakukan augmentasi citra dengan rotasi antara  $0^\circ - 360^\circ$ , perbesar dan perkecil gambar antara 0 – 20 *pixel*, pembalik vertikal dan horizontal yang dilakukan secara acak. Hasil percobaan ini bisa dilihat pada Tabel 3.3

Tabel 3.3 Hasil Percobaan Maria A. Bravo [8]

<b>Preprocessing</b>	<b>VGG-16</b>	<b>Inception V4</b>
Circle RGB	46.3%	44.62%
Square	45.3%	
Color centered	48.3%	45.00%
Gray scale	48.1%	

Saboora Mohammadian dan kawan-kawan [10] menggunakan dataset EyePACS untuk mengklasifikasikan DR ke dalam dua kelas. *Preprocessing* yang dilakukan adalah mengubah ukuran citra agar memiliki ukuran yang sama, warna setiap *pixel* dikurangi dengan nilai rata-rata lokal, lalu citra dipetakan terhadap 50% abu-abu sehingga ketajaman dari citra lebih merata. Pinggiran dari citra retina dipotong untuk menghilangkan efek garis batas. Arsitektur yang dipilih adalah Inception v3 dan Xception yang di-*training* dengan pendekatan *transfer learning*. Model di-*training* sebanyak 200 iterasi dengan nilai *learning rate* 0.0001. Augmentasi yang digunakan adalah pergeseran citra, rotasi citra, pembalikan citra vertikal dan horizontal selama *training*. Saboora membandingkan beberapa

parameter *training* untuk melihat pengaruhnya terhadap performa model yang dihasilkan. Parameter yang dibandingkan adalah optimasi *Adaptive Moment Estimation* (ADAM) dengan SGDM (momentum = 0.9), aktivasi *Exponential Linear Unit* (ELU) dengan ReLU, dan dua blok dengan empat blok yang di-*fine tuning*. Hasil dari percobaan Saboori dapat dilihat pada Tabel 3.4. Hasil terbaik dengan nilai akurasi 87.12% diperoleh dari model Inception v3, optimasi ADAM, dan *fine tuning* sebanyak dua blok.

Tabel 3.4 Hasil Percobaan Saboori Mohammadian [10]

CNN Fine-Tuning	Inception v3	Xception
Unfrozen blocks: 2 Activation function: <b>ReLU</b> Optimizer: <b>SGD</b> With <b>NO</b> data augmentation	0.6048	0.6979
Unfrozen blocks: 2 Activation function: <b>ReLU</b> Optimizer: <b>SGD</b> With data augmentation	0.8074	0.786
Unfrozen blocks: 2 Activation function: <b>ELU</b> Optimizer: <b>SGD</b> With data augmentation	0.5341	0.5031
Unfrozen blocks: 2 Activation function: <b>ReLU</b> Optimizer: <b>ADAM</b> With data augmentation	<b>0.8712</b>	0.7449
Unfrozen blocks: 4 Activation function: <b>ReLU</b> Optimizer: <b>ADAM</b> With data augmentation	0.857	0.5742
Unfrozen blocks: 0 (only fully connected layer) Activation function: <b>ReLU</b> Optimizer: <b>ADAM</b> With data augmentation	0.7314	0.6025

Yuping Jiang dan kawan-kawan [33] menggunakan dataset EyePACS untuk mengklasifikasikan DR ke dalam dua kelas. *Preprocessing* yang dilakukan adalah memotong citra citra dengan *bounding box* bujur sangkar lalu diubah ukurannya menjadi  $512 \times 512$  pixel sambil menjaga aspek rasio dengan menambahkan latar hitam. Intensitas warna diproses dengan normalisasi histogram sehingga fungsi distribusi kumulatif tampak selinier mungkin. Dilakukan juga augmentasi citra dengan rotasi secara acak. Setelah *preprocessing* tersebut, citra diskalakan menjadi  $224 \times 224$  untuk di-*training*. Yuping Jiang mengembangkan arsitektur CNN yang



dapat dilihat pada Tabel 3.5. Model di-*training* dengan pendekatan *end to end learning* sebanyak 600 iterasi, *learning rate* 0.01, dan aktivasi ReLU. Regularisasi *dropout* dengan probabilitas 0.5 digunakan untuk mengurangi *overfitting*. Hasil akurasi yang didapatkan adalah 75.7%.

Tabel 3.5 Arsitektur CNN Yuping Jiang [33]

Layer	Name	Type	Output size	kernel/stride
0	Input	Input	96×224×224	—
1	Con1	Convolution	96×224×224	3×3/1/1
2	Con2	Convolution	96×224×224	3×3/1/1
3	Pool1	Pooling	96×112×112	3×3/2
4	Con3	Convolution	128×112×112	3×3/1/1
5	Con4	Convolution	128×112×112	3×3/1/1
6	Pool2	Pooling	128×56×56	3×3/2
7	Con5	Convolution	256×56×56	3×3/1/1
8	Con6	Convolution	256×56×56	3×3/1/1
9	Pool3	Pooling	256×28×28	3×3/2
10	Con7	Convolution	384×28×28	3×3/1/1
11	Pool4	Pooling	384×14×14	3×3/2
12	Con8	Convolution	384×14×14	3×3/1/1
13	Pool5	Pooling	384×7×7	3×3/2
14	Con9	Convolution	256×7×7	3×3/1/1
15	fc1	InnerProduct	256	—
16	fc2	InnerProduct	2	—

Gabriel Garcia dan kawan-kawan [34] menggunakan dataset EyePACS untuk mengklasifikasikan DR ke dalam dua kelas. *Preprocessing* yang dilakukan adalah mengubah ukuran citra sehingga memiliki ukuran yang sama, citra dikurangi dengan nilai rata-rata lokal, lalu dipetakan ke abu-abu. Setelah *preprocessing* tersebut, citra diubah ukurannya menjadi  $256 \times 256$  pixel. Augmentasi yang dilakukan adalah *flipping* citra dan *cropping* 80% citra. Terdapat lima arsitektur CNN yang dicoba oleh Gabriel seperti pada Tabel 3.6. Model<sub>1</sub> dan Model<sub>2</sub> terinspirasi dari AlexNet yang terdiri dari lapisan *convolution*, *max pooling*, dan *fully connected*. Kedua model ini di-*training* dengan pendekatan *end to end learning*, optimasi SGDM (momentum = 0.9) dan menerapkan regularisasi *dropout* dengan nilai probabilitas 0.65. Sedangkan untuk model yang didasari oleh arsitektur VGG-16 di-*training* dengan pendekatan *transfer learning*. Model yang di-*training* dengan distribusi 50/50 artinya dataset dibuat seimbang, sedangkan model yang di-

*training* dengan distribusi *original* artinya model di-*training* dengan dataset yang tidak seimbang. Model distribusi *original* menggunakan *class weight* dengan perbandingan antara sakit dan sehat sebesar 2.74:1. Varian model VGG di-*training* dengan regularisasi *weight decay* sebesar 0.00005. Hasil dari penelitian Gabriel Garcia bisa dilihat pada Tabel 3.7

**Tabel 3.6** Arsitektur yang digunakan oleh Gabriel Garcia [34]

<i>Network</i>	<i>Distribution</i>	<i>Layers</i>	<i>Training mode</i>	<i>Learning rate</i>
Model1	50/50	6	<i>End to end learning</i>	0.01
Model2	50/50	9	<i>End to end learning</i>	0.01
VGG16	50/50	16	<i>Transfer learning</i>	0.0001
VGG16noFC1	50/50	15	<i>Transfer learning</i>	0.0001
VGG16noFC2	Original	15	<i>Transfer learning</i>	0.0001

**Tabel 3.7** Hasil *training* model Gabriel Garcia [34]

<i>Network</i>	<i>Epochs</i>	<i>Accuracy</i>	<i>Sensitivity</i>	<i>Specificity</i>
Model1	45	63.60%	-	-
Model2	91	66.40%	-	-
VGG16	80	74.30%	62%	86%
VGG16noFC1	75	72.70%	68%	77.60%
VGG16noFC2	80	83.68%	54.47%	93.65%

Yi-Wei Chen dan kawan-kawan [3] menggunakan dua dataset yaitu, EyePACS dan Messidor untuk mengklasifikasikan DR ke dalam dua kelas. *Preprocessing* yang mereka lakukan adalah membuang warna latar belakang dengan metode Otsu's *threshold*, meningkatkan citra dengan *unsharp masking*, mengubah ukuran citra menjadi  $224 \times 224$ ,  $448 \times 448$  dan  $672 \times 672$ . Yi-Wei Chen mengembangkan arsitektur CNN yang disebut dengan istilah SI2DRNet-v1 yang bisa dilihat pada Tabel 3.8. Arsitektur ini terdiri atas 15 lapisan *convolution* dan lima lapisan *pooling* yang di-*training* dengan pendekatan *end to end learning* dan *transfer learning*. Pada pendekatan *end to end learning*, model di-*training* sebanyak 80 *epoch* menggunakan optimasi SGDM (momentum = 0.9), *learning rate* mula-mula 0.001 yang dikurangi dengan *step rate decay* setiap 20 *epoch*, gamma 0.5, dan *weigh decay* 0.0005. Pada pendekatan *transfer learning*, SI2DRNET-v1 di-*training* dengan dataset ImageNet terlebih dahulu, lalu di-*fine tuning* dengan dataset EyePACS. Augmentasi citra dilakukan selama *training* dengan pemotongan acak, rotasi acak ( $60^\circ$ ), pembesaran acak (0.2), dan *random shear* (0.2). Hasil akurasi

yang didapati adalah 0.808 pada dataset EyePACS dan 0.912 pada dataset Messidor.

Tabel 3.8 Arsitektur SI2DRNet-v1 [3]

<i>Type</i>	<i>Filters</i>	<i>Size/Stride</i>	<i>Output</i>
Convolution	32	5x5/2	336x336
Convolution	32	3x3	336x336
Max Pooling		3x3/2	168x168
Convolution	64	5x5	168x168
Convolution	64	3x3	168x168
Convolution	64	3x3	168x168
Max Pooling		3x3/2	84x84
Convolution	128	5x5	84x84
Convolution	128	3x3	84x84
Convolution	128	3x3	84x84
Max Pooling		3x3/2	42x42
Convolution	256	5x5	42x42
Convolution	256	3x3	42x42
Convolution	256	3x3	42x42
Max Pooling		3x3/2	21x21
Convolution	512	5x5	21x21
Convolution	512	3x3	21x21
Convolution	512	3x3	21x21
Convolution	5	1x1	21x21
Global Avg. Pooling		21x21	1x1

Xiaoliang Wang dan kawan-kawan [25] menggunakan dataset dari EyePACS yang dipilih hanya 166 citra saja untuk mengklasifikasikan DR ke dalam lima kelas. *Preprocessing* yang dilakukan mengubah ukuran citra menjadi  $227 \times 227$  untuk AlexNet,  $224 \times 224$  untuk VGG-16 dan  $299 \times 299$  untuk Inception v3. Ketiga arsitektur ini di-*training* dengan pendekatan *transfer learning*. *Training* dilakukan dengan metode *5-fold cross validation*, menggunakan optimasi SGDM (momentum = 0.9), fungsi aktivasi ReLU, dan beberapa parameter lain yang berbeda untuk setiap arsitektur seperti pada Tabel 3.9. Hasil akurasi yang diperoleh adalah 37.43% untuk AlexNet, 50.03% untuk VGG-16, dan 63.23% untuk InceptionNet v3.



Tabel 3.9 Hyperparameter yang digunakan oleh Xiaoliang Wang [25]

Hyperparameter	AlexNet	VGG-16	Inception v3
Initial <i>learning rate</i> , $\alpha$	0.0001	0.0001	0.001
<i>Learning rate</i> decay schedule	Stairwise	Stairwise	Exponential
<i>Learning rate</i> decay factor	0.1	0.1	0.16
Mini-batch size	20	20	32
Momentum, $\beta$	0.9	0.9	0.9

Chunyan Lian dan kawan-kawan [9] menggunakan dataset EyePACS untuk mengklasifikasikan DR ke dalam lima kelas. *Preprocessing* yang mereka lakukan adalah mengubah citra menjadi  $256 \times 256$  pixel, meningkatkan warna citra dengan mengurangi warna lokal rata-rata dan membuang warna latar hitam. Pada dataset dilakukan *subsampling* dan augmentasi untuk mengatasi persoalan dataset yang tidak seimbang. Terdapat tiga arsitektur yang digunakan, yaitu AlexNet, ResNet-50, VGG-16. Ketiga arsitektur ini di-*training* dengan pendekatan *transfer learning*, sedangkan bobot pada bagian *classifier* diinisialisasi dengan inisialisasi Xavier. Parameter yang digunakan selama *training* adalah: *learning rate* 0.001 yang berkurang dengan faktor 10 setiap 27 *epoch*, *batch size* 25, optimasi SGDM (momentum = 0.9), dan *weight decay* 0.0005. Hasil akurasi yang diperoleh pada penelitian ini adalah 73.19% untuk AlexNet, 76.41% untuk ResNet-50, dan 79.04% untuk VGG-16 dengan *preprocessing*, 76.01% untuk model VGG-16 tanpa *preprocessing*.

Zhentao Gao dan kawan-kawan [30] menggunakan dataset yang dibangun sendiri dengan total 4.476 citra untuk mengklasifikasikan DR menurut perawatan yang disarankan. DR diklasifikasikan ke dalam empat kelas, yaitu normal, sedang, berat, dan parah. *Preprocessing* yang dilakukan adalah mentransformasi citra, sehingga setiap citra memiliki ukuran dan bentuk yang sama, dan melakukan normalisasi warna. Augmentasi citra selama *training* adalah pembalikan citra horizontal dan vertikal, rotasi citra antara  $[-25^\circ, 25^\circ]$  secara acak, pembesaran citra antara  $[0.85, 1.15]$ , dan distorsi citra secara acak. Terdapat lima arsitektur yang digunakan oleh Zhentao Gao, yaitu ResNet-18, Resnet-101, VGG-19, dan Inception v3, dan arsitektur Inception@4 yang merupakan modifikasi dari Inception v3. Beberapa parameter yang digunakan untuk *training* model adalah: *batch size* 32, optimasi ADAM, *learning rate* 0.00001, *weight decay* 0.2, dan aktivasi ReLU.

Akurasi yang diperoleh adalah 87.61% untuk ResNet-18, 87.26% untuk ResNet-101, 85.50% untuk VGG-19, 88.35% untuk Inception-v3, dan 88.72% untuk Inception@4.

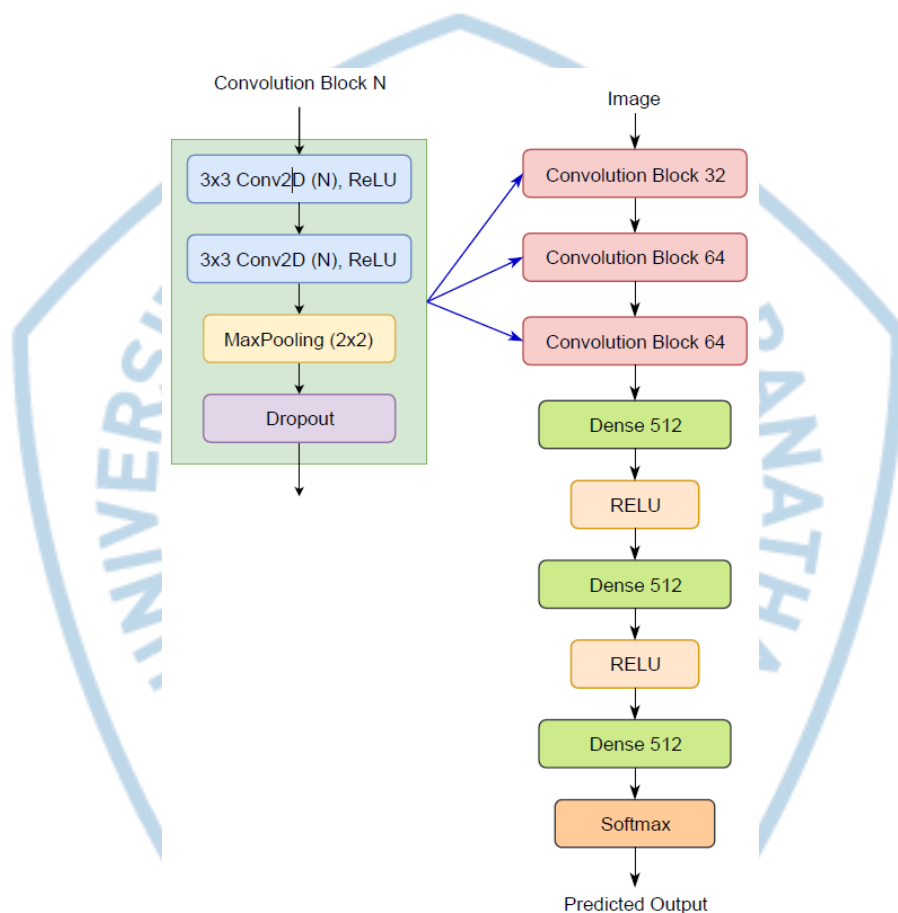
Anjana Umapathy dan kawan-kawan [35] menggunakan dataset dari Karnataka dan STARE untuk mengklasifikasikan DR ke dalam dua kelas. Arsitektur CNN yang dipilih adalah Inception v3. Model di-*training* dengan pendekatan *transfer learning*. Hasil akurasi yang diperoleh adalah 88.8%.

Misgina Tsighe Hagos dan kawan-kawan [36] menggunakan dataset EyePACS untuk mengklasifikasikan DR ke dalam dua kelas. Jumlah dataset yang digunakan terdiri dari 2.500 citra *training* dan 5.000 citra *test*. *Preprocessing* yang mereka lakukan adalah membuang warna latar hitam, mengubah citra menjadi  $300 \times 300$  *pixel*, mengurangi warna rata-rata lokal dari setiap *pixel*. Arsitektur CNN yang digunakan adalah Inception v3 yang di-*training* dengan pendekatan *transfer learning*. Parameter yang dipakai selama *training* adalah: optimasi SGD, *acending learning rate* 0.0005, fungsi *loss cosine*, dan fungsi aktivasi ReLU. Hasil akurasi yang diperoleh adalah 90.9%.

Md Sazzad Hossen dan kawan-kawan [1] menggunakan dataset APTOS-2019 untuk mengklasifikasikan DR menjadi lima kelas. *Preprocessing* yang mereka lakukan adalah mengubah ukuran citra sehingga memiliki radius yang sama, mengurangi citra dengan warna rata-rata lokal yang dipetakan ke 50% *grayscale*, memotong citra menjadi ukuran 90%. Ukuran citra yang digunakan untuk *training* adalah  $224 \times 224$  *pixel*. Arsitektur CNN yang dipilih adalah DenseNet-121 dan di-*training* dengan pendekatan *transfer learning*. Augmentasi yang dilakukan selama *training* adalah pembalikan citra horizontal dan vertikal secara acak dan pembesaran antara 85% - 115% secara acak. Parameter *training* yang digunakan adalah: optimasi ADAM, total *epoch* 12, regularisasi *dropout*, dan fungsi aktivasi ReLU. Hasil yang didapatkan dari dataset validasi adalah akurasi 0.9491, *precision* 0.9598, *recall* 0.9256, F score 0.9395, dan AUC 0.9852.

Quang H. Nguyen dan kawan-kawan [37] menggunakan dataset EyePACS untuk mengklasifikasikan DR ke dalam dua kelas. *Preprocessing* yang mereka lakukan adalah mengubah ukuran citra sehingga memiliki radius yang sama, memotong warna latar belakang citra dan menstandarisasi citra karena memiliki

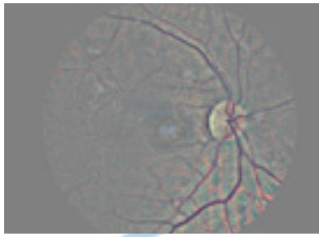

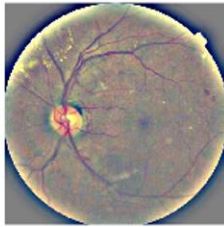
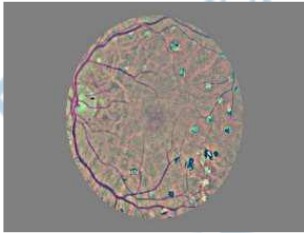
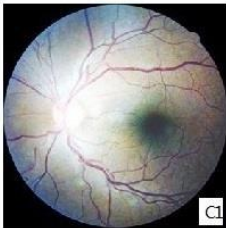
banyak exposure dan variasi cahaya. *Filter* hijau diterapkan untuk mendapatkan kejelasan pada citra yang diikuti dengan konversi citra menjadi *gray scale*. Selama *training*, citra diaugmentasi dengan rotasi dan pencerminan untuk mengatasi dataset yang tidak seimbang. Mereka menggunakan pendekatan *transfer learning* dengan arsitektur CNN VGG-16, VGG-19, dan arsitektur yang dikembangkan sendiri seperti pada Gambar 3.2. Hasil yang diperoleh adalah akurasi 71% untuk VGG-16, 73% untuk VGG-19, dan 82% untuk arsitektur CNN yang dikembangkan sendiri.

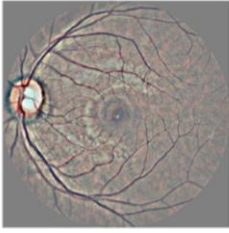
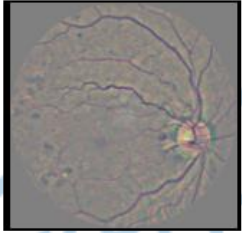





Gambar 3.2 Arsitektur yang digunakan oleh Quang H. Nguyen [37]

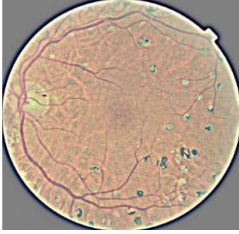
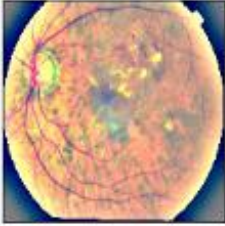
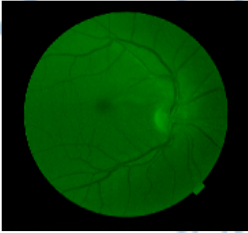
### 3.2 Rangkuman Tinjauan Pustaka

Tabel 3.10 Rangkuman Tinjauan Pustaka

N o.	Peneliti	Dataset	Citra setelah <i>Preprocessing</i>	Klasifikasi	Metode	Arsitektur	ACC %	Kappa%
1	Harry Pratt [31]	EyePACS		5 kelas	End to end	Custom CNN	75	
2	Darshit Doshi [32]	EyePACS		5 kelas	End to end	Custom CNN		39.96
3	Maria A. Bravo [8]	EyePACS		5 kelas	Transfer learning	VGG – 16 Inception v4	48.3 45.0	
4	Saboora Mohamadian [10]	EyePACS		2 kelas	Transfer learning	Inception v3 Xception	87.12 74.49	
5	Yuping Jiang [33]	EyePACS		2 kelas	End to end	Custom CNN	75.7	

N o.	Peneliti	Dataset	Citra setelah <i>Preprocessing</i>	Klasifikasi	Metode	Arsitektur	ACC %	Kappa%
6	Gabriel Garcia [34]	EyePACS		2 kelas	Transfer learning	VGGnet oFC2	83.68	
7	Yi-Wei Chen [3]	EyePACS		5 kelas	End to end	SI2DR Net-v1	80.8	
		Messidor					91.2	
8	Xiaoliang Wang [25]	EyePACS		5 kelas	Transfer learning	AlexNet	37.43	
						VGG-16	50.03	
						Inception-v3	63.23	
9	Chunyan Lian [9]	EyePACS		5 kelas	Transfer learning	AlexNet	73.19	
						ResNet-50	76.41	
						VGG-16	79.04	
10	Zhentao Gao [30]	Custom dataset		4 kelas	Transfer learning	ResNet-18	87.61	
						ResNet-101	87.26	
						VGG-19	85.50	
						Inception-v3	88.35	
						Inception@4	88.72	
11	Anjana Umapathy [35]	Karnataka & STARE		2 kelas	Transfer learning	Inception-v3	88.8	



N o.	Peneliti	Dataset	Citra setelah <i>Preprocessing</i>	Klasifikasi	Metode	Arsitektur	ACC %	Kappa%
12	Misgina Tsighe [36]	EyePACS		2 kelas	<i>Transfer learning</i>	Inception-v3	90.9	
13	Md Sazzad Hossen [1]	APTOS 2019		5 kelas	<i>Transfer learning</i>	DenseNet-121	94.91	
14	Quang H. Nguyen [37]	EyePACS		2 kelas	<i>Transfer learning</i>	VGG-16	71	
						VGG-19	73	
						Custom CNN	82	