

PERANCANGAN IDENTIFIKASI PENYAKIT PADA RETINA BERBASIS PENGOLAHAN CITRA OPTICAL COHERENCE TOMOGRAPHY (OCT) MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)

IDENTIFICATION OF DISEASES IN RETINA BASED ON OPTICAL COHERENCE TOMOGRAPHY (OCT) IMAGE USING CONVOLUTIONAL NEURALNETWORK (CNN)

Dyah Retno Mutia¹, N Kumalasari Caecar Pratiwi², Yunendah Nur Fu'adah³

^{1,2,3} Universitas Telkom, Bandung

dyahretnomutia@telkomuniversity.ac.id¹, caecarnkcp@telkomuniversity.ac.id²,

yunendah@telkomuniversity.ac.id³

ABSTRAK

Penglihatan adalah salah satu indra manusia yang paling penting, kekurangan yang dapat mempengaruhi produktifitas dan kemandirian seseorang. Penyakit retina mempengaruhi jutaan orang dan dapat mengakibatkan kerugian penglihatan jika penyakit tidak didiagnosis dan diobati secara dini. Peningkatan ini menyebabkan diperlukan alat bantu bagi para tenaga medis untuk mendeteksi gejala-gejala yang ditimbulkan akibat penyakit ini. Proses identifikasi dan klasifikasi citra hasil *Optical Coherence Tomography* (OCT) yang akurat membutuhkan tenaga ahli dan sistem deteksi yang mumpuni. Pada penelitian ini dilakukan identifikasi penyakit pada retina berbasis citra *Optical Coherence Tomography* (OCT) menggunakan metode *Convolutional Neural network* (CNN) yang mampu melakukan klasifikasi empat kelas penyakit pada retina yaitu pada kondisi CNV, DME, *Drusen* dan Normal. Parameter sistem yang terbaik di dapatkan dengan output channel 8,16,32,64,128, 5 jumlah *hidden layer*, menggunakan Adam *optimizer*, *learning rate* 0,001 dan *batch size* 32. Hasil yang diperoleh pada kondisi optimal didapatkan dari nilai akurasi, presisi, *recall*, dan *f1-score* yaitu masing-masing 87%, 86,75%, 87,5%, dan 87,25%.

Kata Kunci : Penyakit Pada Retina, CNN, OCT

ABSTRACT

Sight is one of the most important human senses, the lack of which can affect a person's productivity and independence. Retinal disease affects people's illnesses and can result in impaired vision if not diagnosed and treated early. This increase causes the necessary tools for medical personnel to detect the symptoms caused by this disease. The process of knowing and classifying an accurate Optical Coherence Tomography (OCT) image requires experts and a capable detection system. In this study, a retinal disease method based on Optical Coherence Tomography (OCT) images using Convolutional Neural Network (CNN) was carried out which was able to classify four classes of retinal diseases, namely CNV, DME, Drusen and Normal conditions. The best system parameters were obtained with output channels 8,16,32,64,128, 5 hidden layers, using Adam optimizer, learning rate 0.001 and batch size 32. The results obtained under optimal conditions were obtained from the values of accuracy, precision, recall, and f1-scores are 87%, 86.75%, 87.5%, and 87.25% respectively

Keywords : Retina Diseases, CNN, OCT

1. PENDAHULUAN

Penyakit retina adalah selapis tipis sel yang terletak pada bagian belakang dari bola mata. Beberapa penyakit retina yaitu *Choroidal Neovascularization* (CNV), *Diabetic Macular Edema* (DME), *Drusen* yang dapat di deteksi dari gambar hasil scan OCT retina. *Optical Coherence Tomography* (OCT) adalah teknik pencitraan yang menggunakan cahaya untuk menangkap gambar resolusi tinggi dari jaringan biologis. *Optical Coherence Tomography* (OCT) dapat memberikan gambaran dan informasi yang detail mengenai retina. *Convolutional Neural Network* (CNN) adalah salah satu jenis jaringan saraf yang dapat mengidentifikasi fitur struktural. Struktur CNN terdiri dari input, proses ekstraksi fitur, proses klasifikasi dan output. Proses ekstraksi dalam CNN terdiri dari beberapa lapisan tersembunyi atau *hidden*

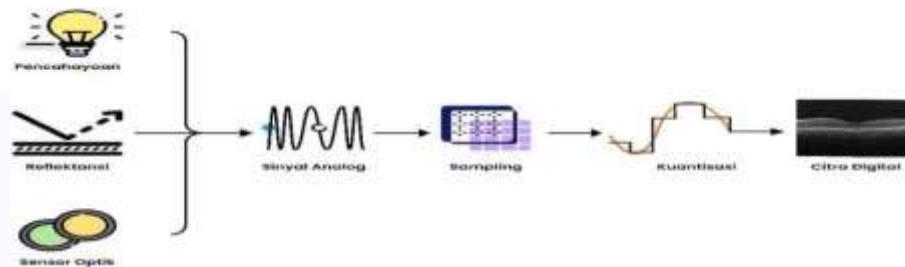
layer, yaitu lapisan *konvolusi*, fungsi aktivasi (ReLU), dan *pooling* [1][2]. Terdapat beberapa penelitian [3][4][5] yang mengembangkan sistem identifikasi penyakit retina menggunakan CNN yang bertujuan untuk mendapatkan parameter yang sesuai.

Dari beberapa penelitian terdahulu yang dibahas diatas, terlihat bahwa CNN cukup handal dalam melakukan identifikasi penyakit retina, sehingga dalam penelitian kali ini akan dibangun sebuah sistem yang mampu melakukan klasifikasi empat kelas penyakit retina yaitu kondisi penyakit CNV, DME, *Drusen*, dan kondisi Normal. Penelitian ini menggunakan data citra yang diambil dari www.kaggle.com, di mana peneliti menguji beberapa parameter berupa pengaruh jumlah *hidden layer*, *optimizer*, *learning rate* dan *Batch Size* terhadap tingkat performansi sistem.

2. DASAR TEORI

2.1. Pengolahan Citra Digital

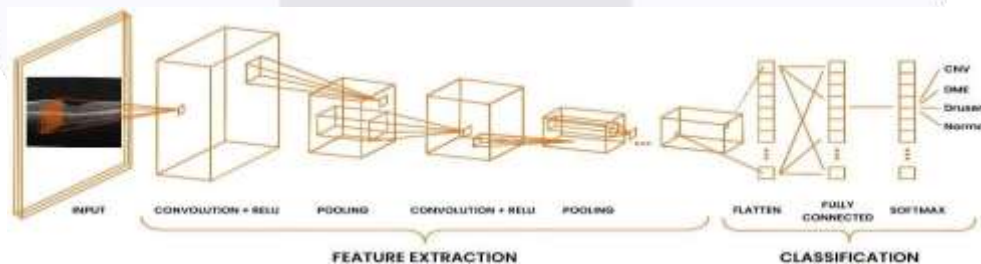
Pengolahan citra digital memiliki tiga komponen dasar untuk membentuk sebuah persepsi terhadap sebuah tampilan yaitu sumber cahaya, objek dan sensor. Pembentukan sebuah citra digital mengalami dua proses utama yaitu pembentukan citra yang dapat di lihat pada gambar 2.1 dibawah. Secara umum format warna terbagi menjadi dua bagian yaitu *grayscale* dan *colored*. Citra dengan format warna *grayscale* pada umumnya mengandung informasi 0 - 255 bits (rentang antara warna putih dan hitam) per piksel. Untuk citra berwarna terdapat beberapa format diantaranya *Red Green Blue* (RGB), *Cyan Magenta Yellow Black* (CMYK), dan *Illumination Blue Red* (YCbCr). Semakin banyak Komponen Warna maka akan semakin banyak informasi yang terkandung. Sebuah citra RGB mengandung 3 x (0-255) bits per pikselnya [6].



Gambar 2. 1 Proses Citra Digital

2.2. Convolution Neural Network

Convolution Neural Network adalah arsitektur yang mampu mengenali informasi prediktif suatu objek (gambar, teks, potongan suara, dsb) walaupun objek tersebut dapat diposisikan dimana saja pada input. Kontribusi CCN yaitu pada *pooling layer*. *Convolution* bekerja dengan prinsip *sliding window* dan *weight sharing* (mengurangi kompleksitas perhitungan). *Pooling layer* berguna untuk merangkul informasi informatif yang dihasilkan oleh suatu *convolution* (mengurangi dimensi). Pada ujung akhir CNN, kita lewatkan satu vektor hasil beberapa operasi *convolution* dan *pooling* pada *multilayer perceptron* (*feed-forward neural network*), dikenal juga sebagai *fully connected layer*. Berikut arsitektur CNN dapat dilihat pada gambar 2.2.



Gambar 2. 2 Arsitektur CNN

2.3. Confusion Matrix

Confusion Matrix merupakan alat pengukuran yang dapat digunakan untuk menghitung kinerja atau tingkat kebenaran dari proses klasifikasi. *Confusion Matrix* dapat menganalisa seberapa baik

classifier untuk mengenali dari tiap kelas – kelas yang berbeda. Berikut merupakan tabel *confusion Matrix* yang dapat dilihat pada Table 3.1

Tabel 3. 1 Proses Confusion Matrix

		Predicted Number			
		Class 1	Class 2	...	Class n
Actual Number	Class 1	x_{11}	x_{12}	...	x_{1n}
	Class 2	x_{21}	x_{22}	...	x_{2n}
	⋮	⋮	⋮	⋮	⋮
	Class	x_{n1}	x_{n2}	...	x_{nn}

Berikut formulasi *confusion matrix multi-class* dijabarkan dengan persamaan dibawah ini :

$$TFN_i = \sum_{j=1, j \neq i}^n x_{ij} \quad (3.1)$$

$$TFP_i = \sum_{j=1, j \neq i}^n x_{ji} \quad (3.2)$$

$$TTN_i = \sum_{j=1, j \neq i}^n \sum_{k=1, k \neq i}^n x_{jk} \quad (3.3)$$

$$TTP_{all} = \sum_{j=1}^n x_{jj} \quad (3.4)$$

Keterangan; *Total of True Positive* (TTP), *Total of True Negative* (TTN), *Total of False Positive* (TFP), *Total of False Negative* (TFN).

Persamaan untuk akurasi dijabarkan dengan persamaan 3.5 di bawah ini :

$$Akurasi = \frac{TTP_{all} + TTN_{all}}{Total\ Number\ of\ Testing\ Entries} \quad (3.5)$$

Persamaan untuk presisi dijabarkan dengan persamaan 3.6 di bawah ini :

$$Precision = \frac{TTP_{all}}{TTP_{all} + TFP_i} \quad (3.6)$$

Persamaan untuk *recall* dijabarkan dengan persamaan 3.7 di bawah ini :

$$Recall = \frac{TTP_{all}}{TTP_{all} + TFN_i} \quad (3.7)$$

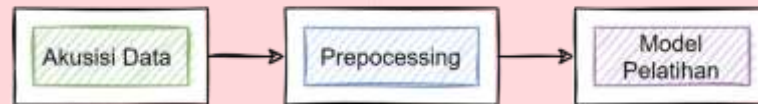
Persamaan untuk *f1-score* dijabarkan dengan persamaan 3.8 di bawah ini :

$$f1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.8)$$

3. PERANCANGAN SISTEM

3.2 Proses Penyisipan dan Ekstraksi

Proses Pada penelitian ini akan dirancang suatu model sistem Pengolahan Citra Digital dengan proses klasifikasi citra menggunakan metode CNN. Berikut blok sistem pada gambar 3.1.



Gambar 3. 1 Blok Sistem

3.2.1 Akusisi

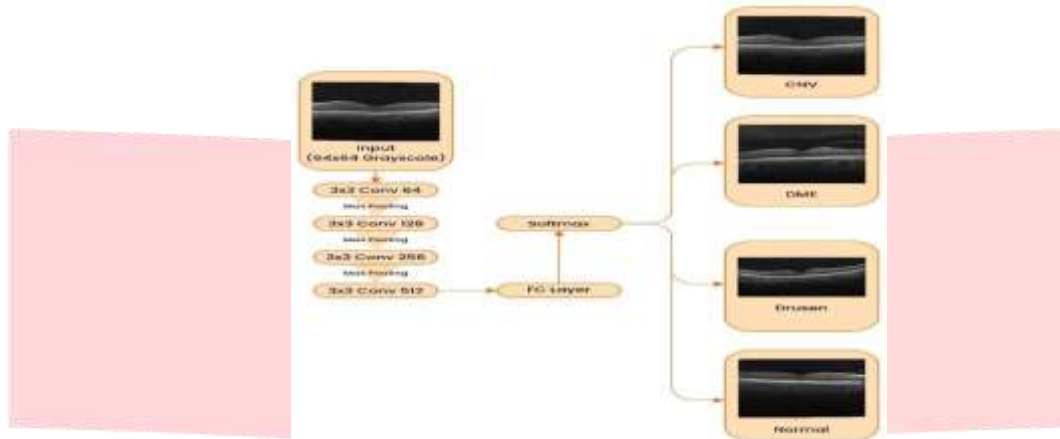
Dalam tugas akhir ini citra yang digunakan adalah citra *Optical Coherence Tomography* (OCT), yaitu hasil *scan* retina. Citra OCT ini di peroleh dari unduhan pada penyedia data biomedis yang beralamat di <http://www.kaggle.com>. Selanjutnya adalah menentukan data yang di perlukan untuk digunakan sebagai data *training* dan data *testing*. Data yang diperoleh untuk nantinya diolah sebanyak 984 data citra OCT. 75% dari keseluruhan data (738 data) dialokasikan untuk *dataset* pada proses *training*. 246 data OCT atau 25% dari keseluruhan data digunakan dalam proses pengujian. Format citra digital yang digunakan adalah *.jpg yang memiliki dimensi 768×496 Pixel.

3.2.2 Preprocessing

Preprocessing pada penelitian ini hanya melewati satu tahap yaitu *resize* (megubah resolusi) pada citra retina dari ukuran sebelumnya, dimana ukuran citra yang tidak sama diubah menjadi 64×64 pixel. *Preprocessing* bertujuan agar data citra lebih mudah untuk diproses pada langkah selanjutnya Model pelatihan.

3.2.3 Model Pelatihan

Proses ekstraksi ciri (*feature extraction*) dan klasifikasi (*classification*) dari tiap citra dengan menggunakan metode CNN. Dengan metode CNN hasil yang diharapkan adalah sistem mampu mengklasifikasi tahapan penyakit Retina. Pada proses model pelatihan akan diterapkan model arsitektur pelatihan CNN Seperti pada Gambar 3.2. Pada umumnya arsitektur CNN memiliki dua proses utama, yaitu proses *feature extraction* dan klasifikasi. Arsitektur CNN memiliki beberapa operasi layer diantaranya yaitu, *Convolutional Layer*, Aktifasi ReLU, *Pooling Layer*, *Fully Connected Layer*, dan diakhiri dengan Aktifasi *Softmax*. Pada input gambar menggunakan citra OCT dengan ukuran $64 \times 64 \times 1$. Angka 1 yang dimaksud adalah dimensi warna karena hasil scan citra memiliki 1 channel yaitu *grayscale*. Tahap selanjutnya, input citra akan masuk ke dalam proses konvolusi, aktifasi ReLU, dan *pooling* yang berada pada tahap *feature learning*. Jumlah proses konvolusi yang akan dilakukan pada penelitian ini diawali dengan 3 lapisan konvolusi, 3 lapisan *fully connected*, dan 1 aktifasi *softmax*. Setiap *convolution layer* memiliki jumlah filter yang sama (3×3). Jumlah channel pada masing-masing *convolution layer* diawali dengan 2, 4, dan 8. Selanjutnya akan dilakukan proses yang terakhir yaitu proses *classification*. Proses ini dimulai dengan tahap perampingan (*flatten*), yang dimaksud dengan tahap ini yaitu mengubah *feature map* hasil *pooling layer* kedalam sebuah vektor. Tahap tersebut bisa disebut juga dengan *fully connected layer*. Proses terakhir yaitu proses aktifasi *softmax*, untuk mengklasifikasikan nilai dari kelas-kelas citra pada *hidden layer*. Hasil yang diharapkan dari proses pada Gambar 3.2 adalah arsitektur CNN mampu mengklasifikasi tahap- tahap penyakit Retina.



Gambar 3. 2 Diagram Alur Pelatihan

4. ANALISIS

4.1. Skenario Pengaruh Jumlah *Hidden Layer*

Pada skenario pengujian pertama dengan melihat pengaruh terhadap jumlah *hidden layer*. Pengujian dilakukan dengan jumlah *hidden layer* sebanyak 3, 4, 5 dan 6 dengan parameter pengujian lainnya sebagai acuan yaitu sesuai dengan tabel 4.1. Hasil dari skenario pertama didapatkan hasil pada tabel 4.1 berikut

Tabel 4.1 Hasil Skenario Pengujian Jumlah Hidden layer

<i>Hidden layers</i>	Output Channel	Performansi				
		Akurasi	Precision	Recall	F1 Score	Loss
3	8, 16, 32	74%	74,25%	73,75%	73,5%	1,8428
4	8, 16, 32, 64	78%	78,75%	77,75%	78%	1,4828
5	8, 16, 32, 64, 128	87%	86,75%	87,5%	87,25%	0,5749
6	8, 16, 32, 64, 128, 256	84%	84,25%	85,25%	84,5%	0,8123

Dari tabel 4.1 dapat disimpulkan bahwa skenario dengan menguji jumlah *hidden layer* mendapatkan performansi sistem terbaiknya pada kondisi jumlah *hidden layer* adalah 5 dengan akurasi sebesar 87% dan *loss* hanya 0,5749.

4.2. Skenario Pengaruh *Algoritma Optimizer*

Dari pengujian sebelumnya, yaitu skenario pengujian jumlah *hidden layer* mendapatkan hasil performansi terbaiknya pada 5 *hidden layer*. Maka pada skenario pengujian kedua yaitu melihat pengaruh *optimizer* (*adam*, *namam*, *SGD* dan *RMSprop*), akan dilakukan pada kondisi 5 *hidden layer* beserta dengan parameter lainnya. Hasil dari skenario kedua didapatkan hasil pada tabel 4.2 berikut

Tabel 4. 2 Hasil Skenario Pemilihan Optimizer

Optimizer	Performansi				
	Akurasi	Precision	Recall	F1 Score	Loss
Adam	87%	86,75%	87,5%	87,25%	0,5749
Nadam	80%	80%	79,75%	79,75%	0,2074
SGD	26%	16,75%	28,75%	16,25%	0,2074
RMSprop	79%	79%	79%	79%	0,5369

Dari tabel 4.2 dapat disimpulkan bahwa skenario pemilihan *optimizer* mendapatkan performansi sistem terbaiknya saat penggunaan *Adam optimizer* dengan akurasi sebesar 87% dan *loss* 0,5749.

4.3. Skenario Pengaruh Nilai *Learning Rate*

Dari pengujian sebelumnya, yaitu skenario pengaruh jumlah *hidden layer* dan skenario pemilihan *optimizer* mendapatkan hasil performansi terbaiknya pada kondisi 5 *hidden layer* dan menggunakan *adam optimizer*. Maka pada skenario ketiga yaitu pengujian pada nilai *learning rate* (0,001, 0,01 dan 0,1) akan dilakukan pada kondisi 5 *hidden layer* dan menggunakan *adam optimizer* beserta dengan parameter lainnya. Hasil dari skenario ketiga didapatkan hasil pada tabel 4.3 berikut

Tabel 4. 3 Hasil Skenario Nilai Learning rate

Learning rate	Parameter				
	Akurasi	Precision	Recall	F1 Score	Loss
0,001	87%	86,75%	87,5%	87,25%	0,5749
0,01	24%	24%	0,25%	39%	1,6923
0,1	18%	4,5%	0,25%	12,25%	1,3895

Tabel 4.3 dapat disimpulkan bahwa skenario mengubah *learning rate* akan menyebabkan sistem mendapatkan performansi terbaiknya pada *learning rate* 0,001 dengan nilai akurasi sebesar 87% dan *loss* 0,5749.

4.4 Skenario Pengaruh *Batch Size*

Dari pengujian sebelumnya, hasil performansi terbaiknya pada kondisi 5 *hidden layer*, *learning rate* 0,001 dan menggunakan *adam optimizer*. Maka pada skenario keempat yaitu pengujian pada *batch size* (32, 64, 128) akan dilakukan pada kondisi 5 *hidden layer* dan menggunakan *adam optimizer* beserta dengan parameter lainnya. Hasil dari skenario keempat dapat dilihat pada tabel 4.4.

Tabel 4. 4 Hasil Skenario Batch size

Batch Size	Parameter				
	Akurasi	Precision	Recall	F1 Score	Loss
32	87%	86,75%	87,5%	87,25%	0,5749
64	83%	84%	82,5%	83%	1,1337
128	77%	77%	77,5%	76,5%	0,9405

Dari tabel 4.4 dapat disimpulkan bahwa skenario mengubah *batch size* akan menyebabkan sistem mendapatkan performansi terbaiknya pada *batch size* 32 dengan nilai akurasi sebesar 87% dan *loss* 0,5749.

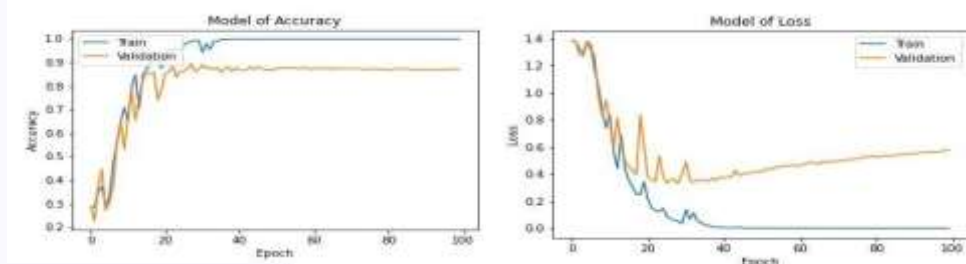
4.4. Hasil Pengujian Terbaik

Dari ketiga skenario pengujian performansi sistem yang telah dilakukan, yaitu skenario pengaruh jumlah *hidden layer*, skenario pemilihan *optimizer* dan skenario nilai *learning rate*, telah didapatkan hasil performansi sistem paling optimal dengan kondisi dan parameternya seperti pada tabel 4.1 berikut:

Tabel 4. 1 Hasil Skenario Pengujian Terbaik

Parameter				Performansi				
Output Channel	Jumlah Hidden layer	Optimizer	Learning rate	Akurasi	Precision	Recall	f1-Score	Loss
8, 16, 32, 64, 128	5	Adam	0,001	87%	86,75%	87,5%	87,25%	0,0340

Dari tabel 4.1 sebagai hasil dari skenario pengujian terbaik, terwujud pula pada grafik performansi akurasi dan *loss*. Grafik tersebut memiliki nilai akurasi tertinggi dan tidak mengalami *overfitting* maupun *underfitting* atau dapat dikatan juga sebagai grafik *good fit* seperti pada gambar 4.1 dibawah ini:



Gambar 4.1 Grafik Performansi akurasi dan loss pengujian terbaik

Karena sistem yang digunakan pada penelitian ini memiliki tingkat akurasi yang tinggi seperti ditunjukkan pada Gambar 4.1 dimana jarak grafik untuk data latih dengan data uji saling berdekatan satu dengan yang lain, maka hal tersebut juga terbukti pada *confusion matrix* seperti gambar 4.2 dibawah ini



Gambar 4. 2 Confusion Matrix pengujian terbaik

Gambar 4.2 menunjukkan bahwa dari total 246 citra data uji, sistem yang digunakan pada penelitian ini mampu mendeteksi sebanyak 246 citra dengan benar sesuai dengan kelasnya.

5. KESIMPULAN

1. Pada penelitian ini berdasarkan keseluruhan skenario pengujian parameter didapatkan saat *hidden layer* berjumlah 5, menggunakan *adam optimizer*, nilai *learning rate* sebesar 0,001 dan *batch size* 32 dengan sistem dapat mencapai akurasi sebesar 87%, *precision* 86,75%, *recall* 87,5%, *f1-score* 87,25%, dan *loss* hanya 0,5749.
2. Sistem klasifikasi penyakit pada retina menggunakan metode *Convolutional Neural Network* (CNN) dapat berfungsi dengan baik. Dimana sistem tersebut mampu mengklasifikasi penyakit retina seperti CNV, DME, Drusen, dan Normal.
3. Proses pengujian pengaruh parameter terhadap performansi sistem pada penelitian ini dilakukan dalam tiga skenario, yaitu dari segi jumlah *hidden layer*, pemilihan *optimizer* dan besaran nilai *learning rate*.

6. REFERENSI

- [1] Maggiori, 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. <https://doi.org/10.1109/TGRS.2016.2612821>
- [2] Katole, A. L., Yellapragada, K. P., Bedi, A. K., Kalra, S. S., & Siva Chaitanya, M. (2015). *Hierarchical Deep Learning Architecture for 10K Objects Classification*. March, 77–93.
- [3] B. K. Triwijoyo *et al.*, “Retina disease classification based on colour fundus images using convolutional neural networks,” *Proc. - 2017 Int. Conf. Innov. Creat. Inf. Technol. Comput. Intell. IoT, ICITech 2017*, vol. 2018-Janua, no. 27, pp. 1–4, 2018, doi: 10.1109/INNOCIT.2017.8319141.
- [4] Mobeen-Ur-Rehman, S. H. Khan, Z. Abbas, and S. M. Danish Rizvi, “Classification of Diabetic Retinopathy Images Based on Customised CNN Architecture,” *Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019*, pp. 244–248, 2019, doi: 10.1109/AICAI.2019.8701231.
- [5] A. Govindaiah, M. A. Hussain, R. T. Smith, and A. Bhuiyan, “Deep convolutional neural network based screening and assessment of age-related macular degeneration from fundus images,” *Proc. - Int. Symp. Biomed. Imaging*, vol. 2018-April, no. Isbi, pp. 1525–1528, 2018, doi: 10.1109/ISBI.2018.8363863.
- [6] Fallis, A. . (2013). Digital Image Processing William Pratt 2007. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).