JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 3, September 2024, Pp. 1442-1455

PENGEMBANGAN MODEL DEEP LEARNING UNTUK DETEKSI RETINOPATI DIABETIK MENGGUNAKAN METODE TRANSFER LEARNING

Yayes Kasnanda Bintang[1)], Helmi Imaduddin[2)]
1. Teknik Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta, Indonesia
2. Teknik Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta, Indonesia

Article Info
Kata Kunci: CNN; Deep learning; Retinopati Diabetik; Transfer Learning

Keywords: *CNN; Deep Learning; Diabetic Retinopathy; Transfer Learning*

Article history:
Received 20 June 2024
Revised 2 July 2024
Accepted 14 August 2024
Available online 1 September 2024

DOI :
https://doi.org/10.29100/jipi.v9i3.5588

Yayes Kasnanda Bintang.
Corresponding Author
E-mail address:
l200190206@student.ums.ac.id

ABSTRAK
Retinopati Diabetik (RD) merupakan komplikasi serius dari diabetes yang dapat menyebabkan kerusakan pada retina dan mengancam penglihatan. Deteksi dini RD sangat penting untuk mencegah kerusakan mata yang lebih lanjut. Dalam usaha untuk meningkatkan deteksi dini ini, teknologi *deep learning*, khususnya metode *CNN*, telah digunakan secara luas. Penelitian ini bertujuan untuk mengimplementasikan dan membandingkan kinerja empat arsitektur *CNN* yang berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*, dalam klasifikasi gambar retina untuk mendeteksi RD. Pertama, dataset gambar retina dibagi menjadi kategori yang terinfeksi RD dan yang tidak. Kemudian, model *CNN* dikembangkan dan dilatih menggunakan data latih untuk mengklasifikasikan gambar. Penggunaan teknik augmentasi data membantu meningkatkan generalisasi model. Setelah melatih model, pengujian dilakukan menggunakan dataset uji yang terpisah untuk mengevaluasi kinerja masing-masing model. Hasil pengujian menunjukkan bahwa *Xception* dan *DenseNet201* menghasilkan kinerja yang sangat baik dalam mendeteksi RD, dengan akurasi, presisi, *recall*, dan *F1-Score* mencapai 96%. Hasil evaluasi ini menegaskan bahwa teknologi *deep learning*, terutama dalam bentuk *CNN*, memiliki potensi besar dalam mendukung diagnosis medis, khususnya dalam deteksi penyakit mata kompleks seperti RD. Penggunaan model-model ini dapat memberikan manfaat yang signifikan bagi pasien RD, memungkinkan deteksi dini yang lebih efektif dan penanganan yang lebih tepat waktu. Dengan demikian, penelitian ini memberikan kontribusi penting dalam pengembangan solusi otomatis untuk diagnosis RD, yang dapat meningkatkan perawatan kesehatan mata secara keseluruhan.

ABSTRACT
Diabetic Retinopathy (DR) is a serious complication of diabetes that can cause damage to the retina and threaten vision. Early detection of RD is very important to prevent further eye damage. In an effort to improve this early detection, deep learning technology, especially CNN methods, has been widely used. This research aims to implement and compare the performance of four different CNN architectures, namely ResNet152V2, Xception, DenseNet201, and InceptionV3, in retinal image classification for detecting RD. First, the retinal image dataset was divided into RD-infected and non-RD infected categories. Then, a CNN model is developed and trained using the training data to classify images. The use of data augmentation techniques helps improve the generalization of the model. After training the models, testing is performed using separate test datasets to evaluate the performance of each model. Test results show that Xception and DenseNet201 produce excellent performance in detecting RD, with accuracy, precision, recall and F1-Score reaching 96%. The results of this evaluation confirm that deep learning technology, especially in the form of CNNs, has great potential in supporting medical diagnosis, especially in the detection of complex eye diseases such as RD. The use of these models could provide significant benefits for RD patients, enabling more effective early detection and more timely treatment. Thus, this research makes an important contribution to the development of automated solutions for RD diagnosis, which can improve overall eye health care.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 3, September 2024, Pp. 1442-1455

I. PENDAHULUAN

RETINOPATI Diabetik (RD) adalah akibat dari diabetes mellitus yang menyerang retina, struktur sensitif terhadap cahaya di belakang mata. Kerusakan pembuluh darah kecil di retina disebabkan oleh paparan glukosa darah tinggi dalam jangka panjang, yang berpotensi menyebabkan masalah penglihatan [1]. RD dapat berakibat fatal karena seringkali tidak menunjukkan gejala pada tahap awal. Namun, jika tidak ditangani dengan cepat, kondisi ini dapat menyebabkan kerusakan serius pada pembuluh darah di retina, meningkatkan risiko kehilangan penglihatan atau kebutaan permanen. RD merupakan penyebab utama kebutaan global karena dapat menyebabkan perdarahan, pembengkakan, dan pembentukan jaringan parut di retina. Risiko kebutaan ini 25 kali lebih tinggi pada penderita diabetes dan berkaitan dengan durasi penyakit tersebut[2].

Pada tahun 2020, perkiraan jumlah orang dewasa usia 20-79 tahun yang mengalami RD di seluruh dunia mencapai 463 juta jiwa. Data epidemiologi menunjukkan bahwa RD merupakan masalah yang sering terjadi pada orang dewasa, terutama pada mereka yang berusia 40 tahun ke atas. Secara global, terdapat sekitar 95 juta orang atau sebesar 35,4% dari total pasien diabetes mellitus yang mengalami kondisi ini. Peningkatan angka penderita diabetes mellitus secara global juga berdampak pada peningkatan insidensi RD [3]. Diperkirakan bahwa sepertiga dari mereka yang mengalami RD menghadapi risiko kehilangan penglihatan. Prevalensi kebutaan global mencapai 1,5 miliar dan sebanyak 0,4 juta kasus dilaporkan disebabkan oleh RD [4]. Data terbaru yang dirilis oleh Kementerian Kesehatan Republik Indonesia (Kemenkes RI) mengindikasikan bahwa pada tahun 2023, sebanyak 16,7% dari individu yang menderita diabetes di Indonesia mengalami RD menunjukkan bahwa hal ini merupakan masalah serius yang perlu mendapatkan perhatian khusus [5].

Pengembangan teknologi *deep learning* telah membawa perubahan revolusioner dalam dunia kesehatan dengan memungkinkan deteksi dini dan prediksi penyakit secara akurat. Salah satu aplikasi yang sangat penting adalah dalam prediksi gejala RD. Pengembangan teknologi *deep learning* untuk prediksi RD mempunyai potensi yang besar dalam meningkatkan akses terhadap pelayanan kesehatan, terutama bagi populasi yang tinggal di wilayah terpencil. Melalui deteksi dini dan terapi yang tepat, risiko komplikasi yang terkait dengan RD dapat diminimalisir, yang tujuan utamanya dapat meningkatkan kualitas hidup bagi individu yang menderita diabetes [6]. Implementasi *deep learning* dan pengembangannya telah membuka peluang baru dalam diagnosis medis dan pengobatan penyakit. *Deep learning* telah terbukti sangat efektif dalam menganalisis gambar medis, termasuk citra fundus untuk mendeteksi dan mengklasifikasikan kondisi medis seperti RD.

Secara umum *deep learning* memiliki kelebihan yang membuat teknologi ini sangat kuat, seperti *deep learning* dapat belajar dari data dalam jumlah besar dan kompleks tanpa diprogram secara eksplisit, memungkinkan identifikasi pola dan hubungan yang rumit. Kemudian dalam beberapa tugas, *deep learning* telah mencapai akurasi yang melebihi kemampuan manusia, seperti pengenalan gambar dan terjemahan bahasa. Model *deep learning* dapat digeneralisasi ke data baru yang tidak dilihat selama pelatihan, sehingga memungkinkan aplikasi yang luas. *Deep learning* juga dapat dijalankan pada *hardware* yang kuat, sehingga memungkinkan pemrosesan data dalam jumlah besar secara efisien. Dari banyaknya kelebihan *deep learning* ternyata juga memiliki keterbatasan seperti *deep learning* membutuhkan data dalam jumlah besar untuk pelatihan, yang mungkin tidak selalu tersedia. Kemudian performa deep learning sangat bergantung pada kualitas data pelatihan. Data yang bias atau bising dapat menghasilkan model yang tidak akurat. Model *deep learning* seringkali dianggap sebagai "kotak hitam" karena sulit untuk memahami bagaimana mereka membuat keputusan. Hal ini dapat menimbulkan masalah interpretasi dan kepercayaan. Pelatihan model *deep learning* seringkali membutuhkan sumber daya komputasi yang signifikan, membuatnya tidak dapat diakses bagi semua orang. *Deep learning* adalah teknologi yang kuat dengan potensi besar untuk meningkatkan berbagai bidang, termasuk deteksi dini RD. Namun, penting untuk memahami kekuatan dan keterbatasannya. Dengan penelitian dan pengembangan yang berkelanjutan, *deep learning* dapat mengatasi keterbatasannya dan menjadi alat yang lebih bermanfaat untuk memecahkan masalah yang kompleks.

Dalam konteks deteksi RD, *Convolutional Neural Network (CNN)* menjadi salah satu pilihan utama. *CNN* merupakan salah satu teknik *deep learning* yang efektif dalam menganalisis gambar, terutama dalam hal deteksi dan klasifikasi objek. *CNN* memiliki kemampuan untuk mengekstrak fitur-fitur penting dari gambar secara otomatis, sehingga cocok digunakan dalam kasus diagnosis RD di mana deteksi dini sangat penting. Dalam konteks penelitian ini, penggunaan *CNN* dan *Hybrid Deep Convolutional Neural Network*s bertujuan untuk meningkatkan akurasi klasifikasi RD melalui ekstraksi fitur yang lebih efektif dari gambar mata fundus. Dalam konteks RD, penggunaan *CNN* bertujuan untuk meningkatkan akurasi diagnosis, memungkinkan deteksi dini penyakit, dan memberikan solusi yang lebih efisien dalam penanganan kasus RD [7].

Pada penelitian terdahulu, metode *CNN* telah berhasil digunakan sebagai pendekatan yang efektif dalam deteksi RD melalui analisis citra fundus. Dengan kemampuannya dalam mengekstraksi fitur kompleks dari gambar medis, *CNN* telah terbukti mampu mengidentifikasi tanda-tanda awal kerusakan pada pembuluh darah di retina dengan

tingkat akurasi yang tinggi. Penelitian ini memberikan landasan penting bagi penggunaan *CNN* dalam upaya deteksi dini dan manajemen penyakit RD. Menurut penelitian [8] yang menggunakan *Deep Learning CNN* dengan arsitektur *VGG16* untuk melatih model klasifikasi RD pada *Optical Coherence Tomography Angiography (OCTA).* Proses *transfer learning* diimplementasikan untuk melatih kembali *CNN* untuk klasifikasi RD. Hasil penelitian menunjukkan bahwa model klasifikasi RD yang paling baik dicapai dengan melatih kembali 9 *layer* terakhir dari *CNN* arsitektur *VGG16*. Akurasi validasi silang untuk membedakan antara mata sehat, mata dengan diabetes tanpa RD dan mata dengan RD adalah 87,27%, dengan sensitivitas 83,76% dan spesifisitas 90,82%. Selain itu, GUI platform juga dikembangkan pada penelitian ini untuk memungkinkan validasi yang lebih mudah dari metode untuk skrining diagnosis RD di lingkungan klinis. Penelitian lain dilakukan oleh [9] yang masih menggunakan metode *CNN transfer learning* untuk mendeteksi citra retina dalam konteks diagnosis RD. Penelitian ini mengevaluasi kinerja model *transfer learning ResNet50*, *VGG16*, dan *InceptionV3* pada dataset yang berbeda. Hasilnya menunjukkan bahwa *ResNet50* mendapat akurasi sebesar 90% pada dataset *APTOS 2019 BD*, sedangkan *VGG16* dan *InceptionV3* masing-masing mendapat akurasi sebesar 90% dan 89% pada dataset *E-Ophtha.*

Berdasarkan latar belakang yang disajikan, rumusan masalah yang diajukan meliputi dua aspek utama. Pertama, adalah hasil perbandingan implementasi arsitektur *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3* dalam membuat model deteksi RD. Kedua, adalah evaluasi hasil pengujian yang diperoleh berdasarkan model terbaik yang dihasilkan oleh masing-masing arsitektur *CNN* tersebut. Dengan membandingkan kinerja berbagai arsitektur *CNN* dalam mendeteksi RD dan menganalisis hasil pengujian dari model terbaik, penelitian ini bertujuan untuk memberikan pemahaman yang lebih mendalam tentang keunggulan dan kelemahan dari masing-masing arsitektur dalam mendukung deteksi dini dan manajemen penyakit RD yang lebih baik. Rekomendasi arsitektur *CNN* terbaik diharapkan dapat meningkatkan prognosis serta manajemen penyakit dan mengurangi risiko kehilangan penglihatan melalui *deep learning*. Penelitian ini memperluas pengalaman peneliti dalam teknologi *deep learning* dan memberikan wawasan tentang kinerja arsitektur *CNN* dalam mendiagnosis RD. Dalam konteks medis, penelitian ini berpotensi mengurangi risiko kebutaan melalui deteksi dini. Bagi pengembang penelitian lainnya, studi ini memberikan landasan untuk pengembangan deteksi penyakit mata dengan *deep learning*, mendorong kolaborasi dalam mengatasi tantangan kesehatan mata. Batasan masalah penelitian ini mencakup beberapa aspek utama. Pertama, metode *deep learning* yang digunakan adalah *CNN*, dengan fokus pada implementasi dan perbandingan arsitektur *ResNet152V2*, *Xception*, *DenseNet201* dan *InceptionV3*. Kedua, penelitian ini menggunakan dataset yang berasal dari *Kaggle* yang dapat diakses melalui laman https://www.*kaggle*.com/competitions/aptos2019-blindness-detection/data, dataset ini umumnya digunakan untuk penelitian deteksi RD. Terakhir, *output* kelas pada hasil klasifikasi akan dibagi menjadi dua kategori, yaitu kelas 0 disebut Normal dan kelas 1 disebut RD. Dengan batasan masalah ini, penelitian akan terfokus pada implementasi dan perbandingan arsitektur *CNN* yang berbeda dalam mendeteksi RD menggunakan dataset yang tersedia, dengan klasifikasi hasil berdasarkan kategori tingkat keparahan RD.

II. METODE PENELITIAN

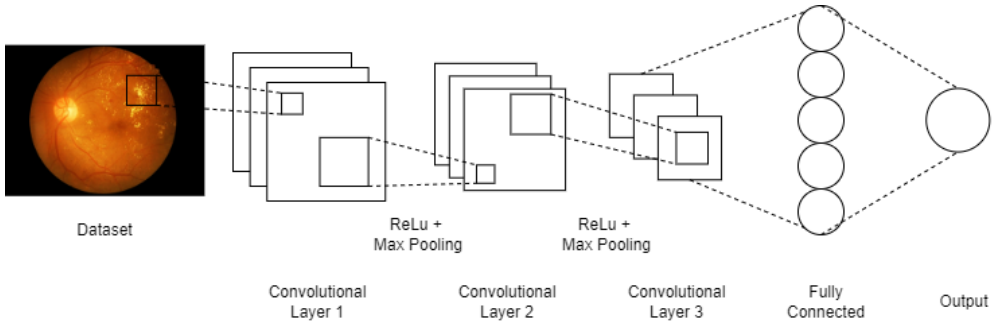
Penelitian ini terbagi menjadi tiga tahapan yang terstruktur dengan baik. Tahap pertama, yaitu tahap *preprocessing,* dimulai dengan pembagian data menjadi data latih, validasi, dan uji untuk memastikan keberagaman dan representasi yang baik dari seluruh dataset. Selanjutnya, dilakukan proses augmentasi pada data latih untuk memperluas variasi data dan meningkatkan generalisasi model. Tahap kedua, yaitu tahap *transfer learning*, melibatkan penggunaan empat model yang berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*. *Transfer learning* memungkinkan pemanfaatan pengetahuan yang sudah ada dalam model tersebut untuk meningkatkan kinerja deteksi RD. Tahap ketiga, yaitu tahap evaluasi yang melibatkan pengukuran tingkat akurasi, presisi, *recall*, dan *F1-Score* untuk menganalisis kinerja model secara menyeluruh. Evaluasi ini memberikan pemahaman yang lebih mendalam tentang keunggulan dan kelemahan dari masing-masing model dalam mendukung deteksi dini dan manajemen penyakit RD.

A. Convolutional Neural Networks

CNN adalah jenis arsitektur jaringan saraf yang dirancang khusus untuk memproses data gambar. *CNN* memiliki lapisan konvolusi yang memungkinkan mereka untuk secara efisien mengekstrak fitur-fitur penting dari gambar. Lapisan konvolusi ini menggunakan *filter* untuk melakukan operasi konvolusi pada gambar *input*, yang membantu dalam menemukan pola visual seperti tepi, tekstur, dan bentuk [10]. Konvolusi adalah proses utama di *CNN* di mana *filter* atau kernel digunakan untuk mengekstrak fitur-fitur lokal dari gambar [7].*CNN* adalah sebuah konsep dalam pembelajaran mendalam yang terkenal dan terinspirasi oleh mekanisme persepsi visual alami pada makhluk hidup [11]. *CNN* telah mengungguli semua pendekatan lain dalam pembelajaran mesin untuk mendeteksi objek visual. Teknik ini didasarkan pada proses persepsi visual yang bawaan pada makhluk hidup dan menjadi salah satu metode pembelajaran mesin paling populer untuk mendeteksi objek visual.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 3, September 2024, Pp. 1442-1455

CNN terdiri dari lapisan konvolusi, *pooling*, dan *fully connected*, yang merupakan komponen penting dalam arsitektur tersebut. Lapisan konvolusi berfungsi untuk mempelajari representasi fitur dari *input* dan menggunakan sejumlah besar kernel konvolusi untuk menghitung berbagai peta fitur [12]. Setiap neuron dalam peta fitur terhubung dengan bagian yang lebih rinci dari lapisan sebelumnya, menghasilkan arsitektur dasar *CNN*. Proses pembentukan peta fitur baru melibatkan konvolusi *input* dengan kernel pembelajaran dan penerapan fungsi aktivasi nonlinier pada keluaran yang dihasilkan [13]. Dari segi struktur, *CNN* merupakan jenis jaringan saraf tiruan *feed-forward* yang sederhana, namun memiliki dua batasan penting yaitu neuron-neuron dalam *filter* yang sama hanya memiliki koneksi ke bagian lokal dari gambar guna menjaga struktur spasial, dan bobotnya dibagikan untuk mengurangi jumlah total parameter model. Sebuah *CNN* terdiri dari tiga komponen utama yaitu lapisan konvolusi yang berfungsi untuk mempelajari fitur-fitur dari data *input*, lapisan *max-pooling* yang bertugas untuk mereduksi dimensi gambar dan secara efektif mengurangi beban komputasi, dan lapisan *fully connected* yang memberikan kemampuan klasifikasi pada jaringan [14]. Ilustrasi mengenai arsitektur *CNN* dapat dilihat pada gambar berikut.



Gambar 1. Arsitektur dasar *CNN*

B. *Convolutional Layer*

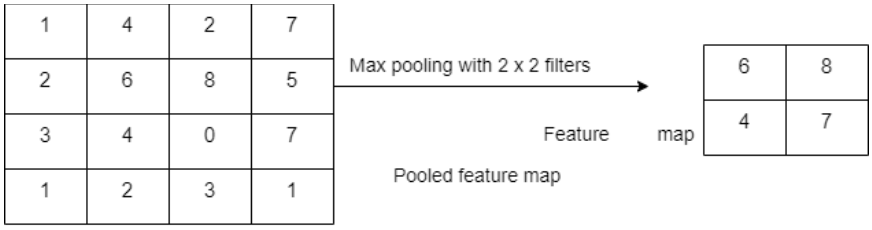
Konvolusi adalah operasi matematis yang penting dalam *CNN*. Pada dasarnya, konvolusi melibatkan penggunaan *filter* atau kernel untuk mengalikan nilai-nilai piksel dari gambar *input* dengan bobot *filter* untuk menghasilkan fitur-fitur yang relevan. Proses konvolusi memungkinkan *CNN* untuk mengekstrak fitur-fitur lokal dari gambar, seperti tepi, sudut, dan tekstur, yang membantu dalam pengenalan pola [7].

$$x_{i,j}^{l} = \sum_{m}\sum_{n} w_{m,n}^{l} \cdot o_{i+m,j+n}^{l-1} + b^{l} \quad (1)$$

Pada *layer* l, persamaan konvolusi untuk matriks dua dimensi melibatkan *input* $o_{i,j}$ dan *filter* $w_{m,n}$ serta bias b, yang menghasilkan $x_{i,j}^{l}$ sesuai dengan persamaan (1). Selanjutnya, $x_{i,j}^{l}$ dapat diaktivasi menggunakan fungsi aktivasi.

C. *Pooling Layer*

Pooling adalah operasi yang digunakan dalam *CNN* untuk mengurangi dimensi data dengan tetap mempertahankan informasi penting. Dua jenis *pooling* yang umum digunakan adalah *max pooling* dan *average pooling*. *Max pooling* memilih nilai maksimum dari setiap jendela *pooling*, sementara *average pooling* menghitung rata-rata nilai dalam jendela *pooling*. *Pooling* membantu mengurangi *overfitting*, mempercepat komputasi, dan membuat representasi fitur lebih *invariant* terhadap pergeseran kecil dalam gambar [7]. *Pooling* adalah langkah yang biasanya mengikuti lapisan konvolusi dalam *CNN*. Tujuannya adalah untuk mengurangi dimensi data dengan menggabungkan informasi dari area yang berdekatan. Proses *pooling* membantu mengurangi *overfitting* dan membuat representasi fitur lebih *invariant* terhadap pergeseran kecil dalam gambar 2[10].



Gambar 2. *Max pooling layer* pada *CNN*

Pooling layer merupakan lapisan yang mengurangi dimensi dari lapisan sebelumnya, yang awalnya berukuran $w * w * f$ menjadi $x * x * f$, di mana x adalah nilai yang lebih kecil dari w. Operasi pada lapisan *pooling* mirip

dengan operasi konvolusi, tetapi tidak melibatkan proses pembelajaran karena tidak ada *filter* yang digunakan. Namun, proses pada lapisan *pooling* dan konvolusi sama-sama melibatkan pergeseran jendela pada *input*. Pada *pooling layer* terdapat beberapa metode yang dapat dilakukan, antara lain *max pooling*, *average pooling*, dan *min pooling*. *Max pooling* mengambil nilai maksimum dari suatu jendela, *average pooling* menghitung nilai rata-rata jendela, dan *min pooling* mengambil nilai minimum dari jendela tersebut. Dalam penelitian ini, dilakukan operasi *max pooling* pada *pooling layer* dengan jendela berukuran 2x2 dan nilai *stride* 2, sebagaimana terlihat dalam gambar 2 di atas [15].

D. *ReLU*

Rectified Linear Unit (ReLU) adalah fungsi aktivasi yang sering digunakan dalam lapisan tersembunyi dari jaringan saraf, termasuk *CNN*. Fungsi *ReLU* didefinisikan sebagai $f(x) = max(0, x)$ yang berarti bahwa nilai *output* adalah nol jika *input* negatif dan sama dengan *input* jika *input* positif. Penggunaan *ReLU* membantu dalam memperkenalkan non-linearitas ke dalam model, mempercepat konvergensi selama pelatihan, dan mengatasi masalah *vanishing gradient*. Fungsi *ReLU* juga lebih efisien dalam komputasi dibandingkan dengan fungsi aktivasi lainnya seperti *sigmoid* atau *tanh* [7]. *ReLU* adalah fungsi aktivasi yang umum digunakan dalam lapisan konvolusi *CNN*. Fungsi *ReLU* mengubah nilai negatif menjadi nol dan mempertahankan nilai positif tanpa perubahan. Ini membantu dalam mempercepat proses pelatihan dan mengatasi masalah *vanishing gradient*. Dengan menggunakan *ReLU*, *CNN* dapat belajar representasi fitur yang lebih baik dan meningkatkan kinerja dalam tugas klasifikasi dan deteksi objek [10].

$$f(x) = \max(0, x) \tag{2}$$

E. *Fully Connected Layer*

Fully Connected Layer (FC layer) merupakan jenis lapisan dalam jaringan saraf dimana setiap neuron di lapisan ini memiliki koneksi ke setiap neuron dalam lapisan sebelumnya dan lapisan sesudahnya. Dalam *FC layer*, tiap neuron menerima *input* dari setiap neuron dalam lapisan sebelumnya dan memberikan *output* ke setiap neuron dalam lapisan berikutnya. Karena hal ini, *FC layer* dikenal sebagai lapisan dengan konektivitas yang paling padat dalam jaringan saraf. Biasanya, *FC layer* ditempatkan di akhir arsitektur jaringan saraf, sebelum lapisan *output*. Fungsinya adalah untuk mengintegrasikan fitur yang telah diekstrak dari lapisan sebelumnya dan melakukan proses klasifikasi atau regresi berdasarkan fitur-fitur tersebut. Penggunaan *FC layer* sangat berguna dalam memahami hubungan yang lebih kompleks antara fitur-fitur yang diekstrak, sehingga memperbaiki kemampuan prediksi jaringan saraf secara keseluruhan. Dalam *CNN*, *FC layer* biasanya digunakan setelah lapisan konvolusi dan *pooling* untuk mengubah representasi spasial dari fitur menjadi representasi linier yang dapat digunakan untuk klasifikasi. Biasanya, *FC layer* diikuti oleh fungsi aktivasi seperti *ReLU* untuk memperkenalkan non-linearitas ke dalam model [7].

$$x_i^l = \sum_m w_m^l \cdot o_{i+m}^{l-1} + b^l \tag{3}$$

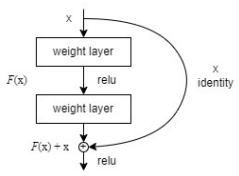
Setiap *node* m dan *input* o dihubungkan oleh bobot w dan bias b, sebagaimana dinyatakan dalam persamaan di atas.

F. *Arsitektur Transfer Learning*

Studi ini menggunakan model *pre-trained ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*. Model *pre-trained* tersebut telah dilatih menggunakan dataset *ImageNet*, yang mencakup fitur mulai dari yang sederhana seperti kecerahan dan batas, hingga fitur yang lebih kompleks dan unik seperti warna dan bentuk [16].

1. *ResNet 152V2*

   Residual Network (ResNet), awalnya dikembangkan untuk menangani dua masalah, seperti masalah hilangnya gradien dan degradasi [17]. Pembelajaran residual digunakan untuk menyelesaikan kedua masalah tersebut. *ResNet152V2* memiliki tiga varian yang berbeda yaitu *ResNet152V218*, *ResNet152V250*, dan *ResNet152V2101* berdasarkan jumlah lapisan dalam jaringan residual. *ResNet152V2* telah berhasil digunakan dalam klasifikasi citra biomedis untuk *transfer learning* seperti pada gambar 3 [18].

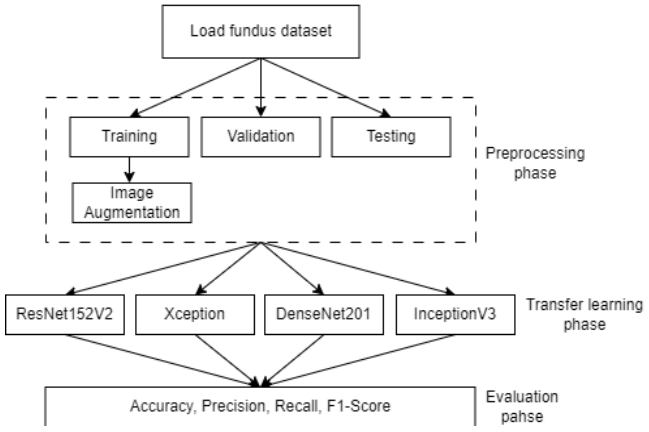


Gambar 3. Arsitektur *ResNet*

2. *Xception*
   Model ini terdiri dari beberapa lapisan yang disebut blok konvolusi terpisah, masing-masing terdiri dari beberapa lapisan konvolusi yang dibagi menjadi dua bagian yaitu konvolusi *depthwise*, dan konvolusi *pointwise*. Hal ini memungkinkan model untuk menangkap informasi pada skala yang lebih kecil dengan jumlah parameter yang lebih rendah, sehingga meningkatkan efisiensi komputasi dan akurasi. *Xception* juga menggunakan teknik normalisasi data yang disebut *batch normalization* untuk mengurangi akurasi tergantung pada skala *input*. Model ini telah terbukti efektif dalam berbagai tugas pembelajaran mesin, termasuk klasifikasi gambar dan pengenalan objek [19].
3. *DenseNet201*
   Model ini terdiri dari lapisan-lapisan yang disebut blok padat atau *dense blocks*, setiap bloknya terdiri dari beberapa lapisan konvolusi yang terhubung langsung ke semua lapisan sebelumnya dalam blok tersebut. Hal ini memungkinkan semua lapisan untuk menerima informasi dari seluruh jaringan, mengurangi kebutuhan akan atribut yang tidak terkait dan meningkatkan transfer informasi di seluruh jaringan. *DenseNet201* memiliki 201 lapisan dan telah terbukti efektif dalam berbagai tugas pembelajaran mesin, termasuk klasifikasi gambar dan pengenalan objek [20]. *DenseNet201* bertujuan untuk memperdalam jaringan pembelajaran mendalam. Perhatian khusus juga diberikan pada efisiensi pelatihan dengan memastikan penggunaan koneksi pendek antar lapisan. Tiap lapisan dalam model terhubung secara langsung ke semua lapisan lainnya. Karena adanya koneksi yang lengkap antar lapisan, aliran informasi antara mereka maksimal. Mekanisme *feed-forward* dari jaringan saraf dijaga dengan memastikan bahwa setiap lapisan menerima *input* dari seluruh lapisan sebelumnya. Peta fitur dari tiap lapisan dikirimkan ke semua lapisan berikutnya. Dalam *DenseNet201*, fitur-fitur digabungkan, berbeda dengan *ResNet152V2* yang menggabungkan fitur melalui penjumlahan. Artinya, lapisan ke-i memperoleh *input* dan terdiri dari peta fitur yang digabungkan dari blok-blok di bawahnya. Peta fitur dari lapisan ke-i diteruskan ke lapisan i yang tersisa, di mana i adalah total jumlah lapisan [21].
4. *InceptionV3*
   Model ini terdiri dari beberapa lapisan yang disebut blok *inception*, setiap bloknya terdiri dari beberapa lapisan konvolusi dengan *filter* berukuran berbeda yang berjalan secara paralel, kemudian disatukan. Hal ini memungkinkan model untuk menangkap informasi pada skala yang berbeda secara simultan, sehingga meningkatkan efisiensi komputasi dan meningkatkan akurasi. *InceptionV3* juga menggunakan teknik normalisasi data yang disebut *bacth normalization* untuk mengurangi akurasi tergantung pada skala *input.* Model ini telah terbukti efektif dalam berbagai tugas pembelajaran mesin, termasuk klasifikasi gambar dan pengenalan objek [22].

G. *Perancangan Pemodelan*

Pada subbab perancangan pemodelan ini, akan dibahas beberapa proses yang menjadi tahapan penting dalam pengembangan model deteksi RD. Tahapan pertama adalah pengumpulan dataset, di mana data yang diperlukan untuk pelatihan dan pengujian model akan dikumpulkan dari sumber yang tersedia. Selanjutnya, *preprocessing* data akan dilakukan, yang meliputi partisi data untuk melatih dan menguji model, penanganan ketidakseimbangan data untuk mengatasi distribusi yang tidak seimbang antar kelas, dan augmentasi data untuk meningkatkan keragaman dan kuantitas data latih. Setelah itu, *transfer learning* akan dilakukan menggunakan empat model yang berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*. Penggunaan empat model yang berbeda ini bertujuan untuk membandingkan kinerja dan efektivitas masing-masing model dalam mendeteksi RD.

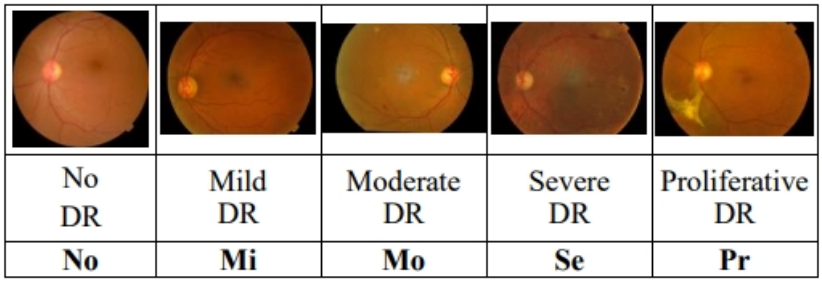


Gambar 4. Tahapan penelitian

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 3, September 2024, Pp. 1442-1455

H. Pengumpulan Data

Penelitian ini menggunakan citra retina dari kumpulan data *APTOS 2019 BD*, sebuah kompetisi terbuka di platform Kaggle [23]. Kumpulan data tersebut terdiri dari 3.662 citra retina yang dikumpulkan dari beragam subjek yang tinggal di wilayah pedesaan India. Citra-citra ini dikumpulkan oleh rumah sakit mata Aravind di India, dengan tujuan mengembangkan model pembelajaran mesin untuk mendeteksi kebutaan secara otomatis tanpa memerlukan pemeriksaan medis. Kumpulan citra ini berupa gambar berwarna Merah-Hijau-Biru (RGB) yang diambil menggunakan teknik fotografi fundus. Setiap sampel kemudian diberi label oleh dokter-dokter terlatih yang mengklasifikasikan tingkat kebutaan seseorang ke dalam lima level yang berbeda yaitu *non-DR*, *Mild*, *Moderate*, *Severe*, dan *Proliverative DR*. Meskipun demikian, tetap dilakukan antisipasi adanya *noise* dalam citra dan label kelas yang sesuai, serta kemungkinan adanya artefak dan ketidakfokusan dalam kumpulan gambar tersebut. Kumpulan citra ini diambil dalam berbagai kondisi dan lingkungan selama periode waktu yang cukup panjang. Gambar 5 menyajikan contoh citra untuk setiap kelas dalam kumpulan data *APTOS 2019 BD* beserta label yang diberikan untuk referensi selanjutnya [24].



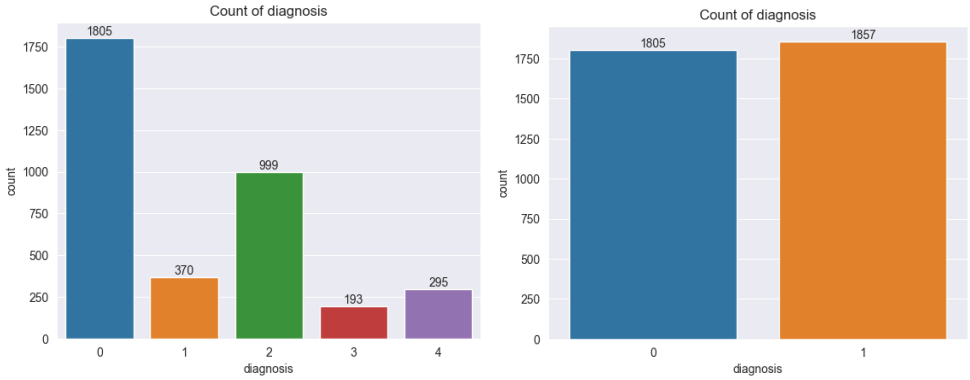
Gambar 5. Sampel kelas RD

I. Data Preprocessing

Pada subbab ini, serangkaian proses dilakukan untuk mempersiapkan data sebelum masuk ke tahap pemodelan. Pertama, dilakukan penanganan ketidakseimbangan data untuk menangani masalah distribusi kelas yang tidak seimbang. Selanjutnya, data dibagi menjadi subset latih, validasi, dan uji menggunakan metode partisi data yang tepat. Proses partisi data ini penting untuk memastikan bahwa model dapat dinilai dan dievaluasi dengan benar. Selain itu, dilakukan juga augmentasi data untuk meningkatkan variasi dataset dan mengurangi risiko *overfitting*.

1. *Imbalanced Data*

   Pada proses penanganan ketidakseimbangan data, fokus utamanya adalah pada situasi di mana kelas mayoritas memiliki jumlah data yang jauh lebih banyak dibandingkan dengan kelas minoritas. Dalam konteks deteksi *RD*, kelas *0 - No DR* seringkali memiliki jumlah data yang signifikan lebih banyak daripada kelas lainnya, seperti kelas *1 - Mild*, *2 - Moderate*, *3 - Severe*, dan *4 - Proliferative DR*. Untuk mengatasi ketidakseimbangan ini, salah satu pendekatan yang dapat dilakukan adalah dengan melakukan peleburan atau penggabungan kelas-kelas minoritas menjadi satu kelas baru. Dalam kasus ini, kelas *1 - Mild*, *2 - Moderate*, *3 - Severe*, dan *4 - Proliferative DR* dapat digabungkan menjadi satu kelas baru yang disebut kelas 1 RD. Dengan demikian, data yang awalnya terbagi ke dalam berbagai kelas RD sekarang digabungkan menjadi dua kelas yaitu kelas 0 untuk non-RD dan kelas 1 untuk RD. Langkah ini diambil untuk memperbaiki keseimbangan antara kelas dalam dataset, sehingga model yang dibangun memiliki kemampuan yang lebih baik untuk mengenali dan membedakan antara kasus RD dan non-RD seperti pada gambar 6.



Gambar 6. Kondisi data awal sebelum dilakukan peleburan kelas (kiri) dan Kondisi data setelah dilakukan peleburan kelas (kanan)

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 3, September 2024, Pp. 1442-1455

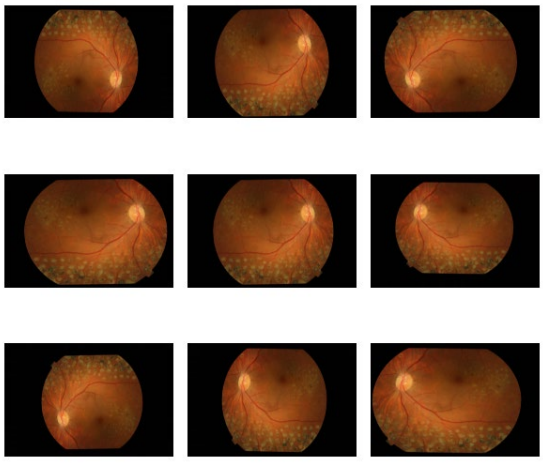
Kondisi awal data menggambarkan jumlah data dalam setiap kategori kelas pada suatu dataset. Dalam kasus ini, terdapat lima kategori kelas yang mewakili tingkat keparahan RD yaitu *No-DR*, *Mild*, *Moderate*, *Severe*, dan *Proliferative DR*. Kondisi data awal pada kelas 0 terdiri dari 1805 citra, kelas 1 terdiri dari 370 citra, kelas 2 terdiri dari 999 citra, dan kelas 3 terdiri dari 193 citra, dan kelas 4 terdiri dari 295 citra seperti yang digambarkan pada gambar 6 di atas. Kemudian setelah dilakukan peleburan data, terjadi penggabungan beberapa kategori kelas menjadi satu kategori kelas baru. Dalam hal ini, kategori kelas 1, 2, 3 , dan 4 digabungkan menjadi satu kategori kelas baru yang disebut RD. Jumlah sampel dalam kategori kelas baru ini adalah hasil penjumlahan dari jumlah sampel dalam kategori kelas yang digabungkan, yaitu Normal berjumlah 1805 atau tanpa perubahan dan RD berjumlah 1857.

2. Data Partisi

Proses partisi data merupakan langkah penting dalam pembuatan model pembelajaran mesin. Pada proses ini, dataset awal dibagi menjadi tiga bagian yang berbeda yaitu data latih, validasi, dan uji. Hasil dari proses ini adalah terbaginya dataset awal menjadi tiga subset yang berbeda antara lain dataset latih, dataset validasi, dan dataset uji. Dataset latih digunakan untuk melatih model, dataset validasi digunakan untuk mengevaluasi performa model selama proses pelatihan, dan dataset uji digunakan untuk mengevaluasi performa akhir model setelah proses pelatihan selesai. Dengan pembagian dataset ini, dapat dilakukan pengembangan dan evaluasi model secara yang lebih terstruktur dan akurat.

3. Augmentasi Data

Proses yang dilakukan selanjutnya adalah konfigurasi untuk augmentasi data gambar menggunakan *ImageDataGenerator* dari *library Keras*. Proses ini bertujuan untuk meningkatkan variasi data latih dengan membuat modifikasi terhadap gambar-gambar yang ada. Berikut merupakan hasil augmentasi data gambar menggunakan *ImageDataGenerator*.



Gambar 7. Sampel proses augmentasi

Parameter yang digunakan pada augmentasi citra antara lain *rescale* untuk mengurangi kompleksitas model dan mempercepat proses pelatihan, *zoom range* yang mengatur rentang *zoom in* dan *zoom out* yang dapat diterapkan pada citra, *shear range* yang mengatur rentang pemutarannya pada citra, *horizontal flip* yang mengatur apakah akan dilakukan pembalikan secara horizontal pada citra, dan *vertical flip* mengatur apakah akan dilakukan pembalikan secara vertikal pada citra.

J. Pemodelan

Pada subbab pemodelan ini, akan menerapkan empat model yang berbeda untuk analisis klasifikasi, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*. Keempat model tersebut telah terbukti efektif dalam berbagai tugas pengenalan gambar dan klasifikasi, serta sering digunakan dalam konteks *transfer learning* untuk memperoleh representasi fitur yang kuat dari citra. Melalui penggunaan empat model yang berbeda ini, bertujuan untuk mengevaluasi kinerja dan kemampuan klasifikasi masing-masing model dalam mendeteksi RD dalam gambar retina. Dengan membandingkan hasil dari keempat model tersebut, agar dapat menentukan model mana yang paling sesuai dan efektif untuk tugas klasifikasi yang dihadapi.

Fine tuning adalah teknik dalam *deep learning* yang melibatkan proses menyesuaikan model jaringan saraf yang telah dilatih sebelumnya pada dataset besar misalnya *ImageNet* ke dataset yang lebih kecil atau tugas spesifik. Dalam konteks *CNN*, *fine tuning* melibatkan jaringan yang telah dilatih pada dataset umum dan menyesuaikannya dengan dataset target dengan tujuan untuk meningkatkan kinerja pada tugas tertentu. Proses *fine tuning* biasanya melibatkan pembekuan beberapa lapisan awal dari jaringan yang mengandung fitur-fitur umum dan hanya melatih lapisan-lapisan yang lebih tinggi untuk tugas spesifik. Dengan *fine tuning*, model dapat memperoleh representasi

fitur yang lebih baik untuk dataset target, meskipun dataset target memiliki ukuran yang lebih kecil. *Fine tuning* merupakan strategi yang efektif dalam *transfer learning* untuk meningkatkan kinerja model pada tugas klasifikasi gambar, deteksi objek, dan tugas lainnya [10].

Teknik *fine tuning* yang digunakan melibatkan *transfer learning* dengan model *pre-trained* pada *ResNet152V2*, *DenseNet201*, *Xception*, dan *InceptionV3*. Penelitian ini menggunakan pendekatan *pre-trained* model *approach* di mana model *pre-trained* tersebut digunakan sebagai titik awal untuk membangun model hybrid *CNN* dengan *ResNet152V2*, *DenseNet201*, *Xception*, dan *InceptionV3* dalam klasifikasi RD. Dengan menggunakan model *pre-trained* tersebut, penelitian ini bermaksud melakukan *fine tuning* pada lapisan-lapisan model untuk menyesuaikan dengan dataset yang digunakan dalam tugas klasifikasi RD. *Fine tuning* dilakukan untuk mengoptimalkan kinerja model dalam memprediksi kehadiran RD dengan akurasi yang diinginkan tanpa mengalami *overfitting* [7].

Dengan demikian, teknik *fine tuning* yang digunakan dalam penelitian ini melibatkan penggunaan model *pre-trained* sebagai titik awal, diikuti dengan penyesuaian lapisan-lapisan model melalui proses *fine tuning* untuk meningkatkan kinerja model dalam tugas klasifikasi RD. Proses *fine tuning* yang dilakukan pada penelitian dijelaskan pada tahapan berikut. Pada tahap awal setelah pengambilan *layer* dari *pre-trained* model, *layer pre-trained* model ini akan dibekukan, hal ini diperlukan agar tidak merusak informasi apa pun yang dikandungnya selama pelatihan berikutnya. Setelah model menyatu dengan data baru dimana itu adalah lapisan yang dibuat, proses *fine tuning* dilakukan dengan mencairkan seluruh atau sebagian model dasar dan melatih ulang seluruh model secara *end-to-end* dengan kecepatan pembelajaran yang sangat rendah. Langkah ini berpotensi akan memberikan peningkatan secara bertahap pada *output* yang dihasilkan. Hal ini juga berpotensi menyebabkan *overfitting*, maka ini hanya perlu dilakukan setelah model dengan lapisan beku dilatih untuk melakukan konvergensi.

Proses *fine tuning* ini memungkinkan kita untuk mengadaptasi model yang sudah ada terhadap data yang baru dengan cara mengubah sebagian kecil dari parameter model yang sudah dilatih sebelumnya. Hal ini dapat membantu meningkatkan kinerja model pada tugas tertentu tanpa harus melatih model dari awal.

K. Pengujian Model

Untuk mengukur kinerja model yang digunakan untuk klasifikasi, digunakan *confusion matrix* atau matriks konfusi. Evaluasi model dilakukan dengan memonitor jumlah positif benar, negatif benar, positif salah, dan negatif salah[25].

Tabel 1. *Confusion matrix*

		Aktual	
		0	1
Prediksi	0	TP	FP
	1	FN	TN

Dari data-data ini, akurasi, presisi, *recall*, dan *F1-Score* dapat dihitung [26].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1\,Score = 2\,x\,\frac{Precision\,x\,Recall}{(Precision + Recall)} \quad (7)$$

Berikut adalah istilah-istilah yang digunakan untuk menggambarkan kinerja model:

1. *True Positive (TP)* adalah data gambar RD positif yang diprediksi dengan benar.
2. *True Negative (TN)* adalah data gambar RD negatif yang diprediksi dengan benar.
3. *False Positive (FP)* adalah data gambar RD negatif yang salah diprediksi sebagai positif.
4. *False Negative (FN)* adalah data gambar RD positif yang salah diprediksi sebagai negatif.

III. HASIL DAN PEMBAHASAN

Pada bab ini, akan dibahas tentang hasil dari proses pemodelan dan pengujian yang telah dilakukan. Subbab yang

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 3, September 2024, Pp. 1442-1455

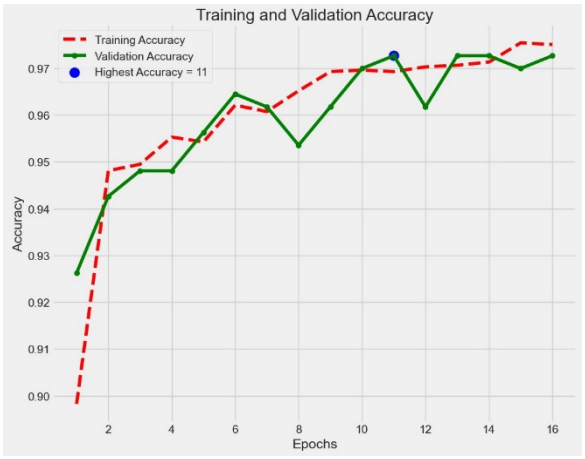
akan dibahas meliputi proses pemodelan menggunakan empat model yang berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*, serta pengujian performa dari masing-masing model. Sebelumnya, data telah dipartisi menjadi tiga bagian, yaitu data pelatihan dengan jumlah 2929, data validasi dengan jumlah 366, dan data pengujian dengan jumlah 367.

A. Pelatihan

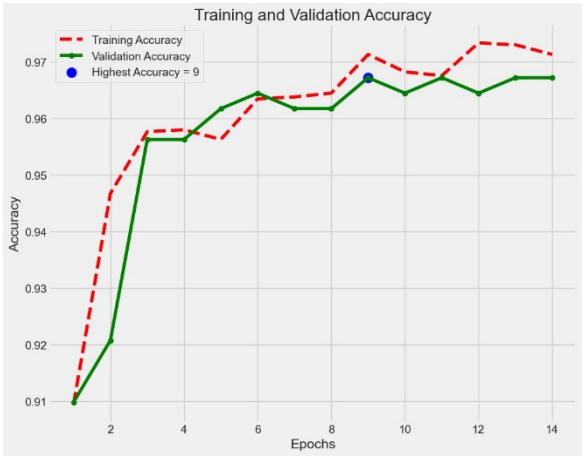
Implementasi empat arsitektur *CNN*, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*, dalam klasifikasi RD dilakukan dengan menggunakan data latih sebanyak 2929 data dan data validasi sejumlah 366 untuk memantau kinerja model selama proses pelatihan. Setiap arsitektur *CNN* dievaluasi dengan memperhatikan akurasi, fungsi kerugian, dan potensi *overfitting*. *Overfitting* adalah masalah umum dalam *machine learning* yang terjadi ketika model terlalu terlatih pada data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data baru. Hal ini dapat mengakibatkan performa model yang buruk pada data yang tidak dilihat selama pelatihan.

Untuk menganalisis *overfitting* dalam model *machine learning*, penting untuk melihat bagaimana performa model berubah pada data pelatihan dan data validasi selama pelatihan. Plot kurva pembelajaran menunjukkan bagaimana metrik performa model berubah dengan jumlah iterasi pelatihan. Jika model *overfitting*, metrik performa pada data pelatihan akan terus meningkat, sementara metrik performa pada data validasi akan mulai menurun pada titik tertentu. Dalam penelitian ini *early stopping* berfungsi untuk menghentikan pelatihan ketika metrik validasi mencapai puncaknya, hal ini dilakukan agar mencegah terjadinya *overfitting*. *Early stopping* adalah teknik yang efektif untuk mencegah *overfitting* dan meningkatkan stabilitas dan generalisasi model *machine learning*. Dengan menganalisis kurva pembelajaran, metrik validasi, dan menggunakan *early stopping*, maka model dapat menggeneralisasi dengan baik pada data baru.

Proses pelatihan dimulai dengan konfigurasi *checkpoint* model untuk menyimpan model dengan akurasi validasi terbaik serta penerapan *early stopping* untuk menghentikan pelatihan jika tidak terjadi peningkatan kinerja dalam beberapa *epoch* berturut-turut. Selanjutnya, model dikompilasi menggunakan optimasi *Adamax* dengan laju pembelajaran 0.0001, fungsi kerugian *BinaryCrossentropy*, dan metrik evaluasi berupa akurasi biner. Evaluasi dilakukan pada setiap *epoch* dengan mencatat nilai kerugian dan akurasi pada data pelatihan dan validasi. Hasil pemodelan dari keempat model arsitektur tersebut dijelaskan pada gambar berikut

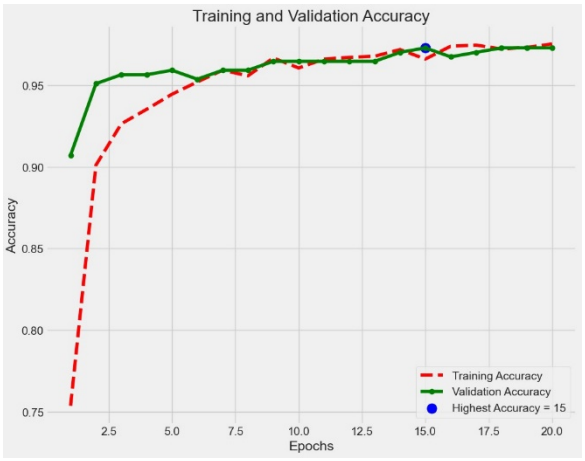


Gambar 9. Hasil *training ResNet152V2*

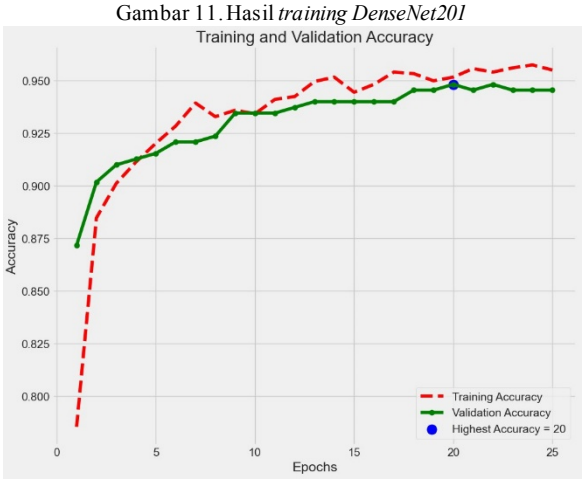


Gambar 10. Hasil *training Xception*

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 3, September 2024, Pp. 1442-1455



Gambar 11. Hasil *training DenseNet201*



Gambar 12. Hasil *training InceptionV3*

Proses pelatihan model *ResNet152V2* dilakukan selama 15 *epoch* dengan iterasi pada data pelatihan dan validasi. Akurasi pada kedua jenis data tersebut meningkat secara bertahap selama proses pelatihan, meskipun prosesnya dihentikan setelah *epoch* ke-15 karena tidak ada peningkatan dalam akurasi validasi setelah *epoch* ke-10. Akurasi model pada data validasi meningkat secara signifikan dari sekitar 93,72% pada *epoch* pertama menjadi sekitar 96,99% pada *epoch* ke-10, dan meskipun tetap stabil setelah itu, tetap tinggi. Kerugian pada model pada kedua jenis data juga menurun secara konsisten, menandakan peningkatan kinerja klasifikasi.

Model *Xception* berhasil mencapai akurasi tertinggi sebesar 96,17% pada *epoch* ke-5 dalam klasifikasi data validasi. Namun, tanpa peningkatan tambahan dalam akurasi setelah itu, proses pelatihan dihentikan lebih awal pada *epoch* ke-10 menggunakan metode *early stopping*. Meskipun tidak ada tanda-tanda *overfitting* yang signifikan, terdapat perbedaan antara akurasi pada data pelatihan dan validasi, menunjukkan potensi penyesuaian lebih lanjut dengan data pelatihan. Meskipun kerugian pada data pelatihan cenderung menurun seiring dengan peningkatan *epoch*, terdapat peningkatan kerugian pada data validasi pada beberapa titik, menandakan kesulitan model dalam generalisasi pada data yang belum pernah dilihat sebelumnya. Penggunaan *early stopping* terbukti efektif dalam menghindari *overfitting*, menghentikan pelatihan pada saat yang tepat saat tidak ada peningkatan akurasi validasi yang signifikan.

Model *DenseNet201* awalnya memiliki akurasi sekitar 76% pada data pelatihan, tetapi meningkat secara signifikan menjadi sekitar 97,4% pada akhir pelatihan, menunjukkan kemampuan model dalam mempelajari pola-pola kompleks dan melakukan klasifikasi dengan akurat. Akurasi pada data validasi juga meningkat dari sekitar 87,9% menjadi sekitar 97,5%, menunjukkan kemampuan generalisasi model yang baik. Meskipun terdapat tanda-tanda *overfitting* pada data pelatihan, tindakan pencegahan seperti penggunaan *EarlyStopping* berhasil menghentikan pelatihan pada saat yang tepat untuk mencegah pembelajaran pola-pola terlalu spesifik pada data pelatihan. Penggunaan optimasi *Adamax* dengan laju pembelajaran 0.0001 dan fungsi kerugian *BinaryCrossentropy* memberikan hasil yang baik dengan mencapai akurasi tinggi pada kedua jenis data.

Dari hasil pemodelan *InceptionV3*, terlihat bahwa model mulai belajar pola-pola penting dalam data validasi sejak *epoch* pertama dengan peningkatan akurasi data validasi dari *-inf* menjadi 0.90164. Kinerja model terus meningkat seiring dengan berjalannya *epoch*, mencapai puncaknya pada *epoch* 19 dengan akurasi data validasi sebesar 0.94536. Pelatihan dihentikan setelah 24 *epoch* karena tidak ada peningkatan akurasi data validasi dalam 5

epoch berturut-turut setelah *epoch* ke-19. Model *InceptionV3* menunjukkan performa yang sangat baik dengan akurasi data validasi mencapai 94,54%. Meskipun terjadi sedikit *overfitting* karena terdapat perbedaan antara akurasi data pelatihan sebesar 95,19% dan akurasi data validasi sebesar 94,54%, namun selisihnya tidak signifikan, menjadikan model masih dapat dianggap baik.

B. Pengujian

Pada subbab pengujian ini, dilakukan evaluasi performa terhadap empat arsitektur jaringan konvolusi yang berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*. Pengukuran performa dilakukan menggunakan metode *confusion matrix* untuk mengukur tingkat akurasi, presisi, *recall*, dan *F1-Score* dari setiap arsitektur. Hasil pengujian akan dipresentasikan melalui *confusion matrix* dan *classification report* untuk masing-masing model. *Confusion matrix* memberikan gambaran tentang kinerja model dalam mengklasifikasikan data uji ke dalam kelas yang benar dan salah, sementara *classification report* memberikan informasi lebih lanjut tentang akurasi, presisi, *recall*, dan *F1-Score* dari setiap kelas. Hasil *confusion matrix* pada pengujian dari keempat model disajikan pada tabel berikut.

Tabel 2. *Confusion matrix* hasil pengujian

Model	TP	TN	FP	FN
ResNet152V2	169	174	18	6
Xception	170	182	10	5
DenseNet201	168	183	9	7
InceptionV3	169	177	15	6

Dari data *confusion matrix* yang disajikan untuk empat model klasifikasi, dapat diamati bahwa masing-masing model memiliki kelebihan dan kelemahan dalam melakukan klasifikasi. Model *Xception* menonjol dengan jumlah positif salah yang paling rendah, menunjukkan kemampuannya dalam menghindari kesalahan mengklasifikasikan kasus negatif sebagai positif. Sementara itu, *DenseNet201* memiliki jumlah negatif benar yang paling tinggi, menandakan kemampuannya dalam mengidentifikasi sebagian besar kasus negatif dengan benar. Namun, keempat model tersebut masih menunjukkan kecenderungan untuk membuat kesalahan klasifikasi, terutama dalam bentuk negatif salah dan positif salah.

Pada pengujian ini, dilakukan evaluasi performa terhadap empat arsitektur model yang berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3*. Dari hasil pengukuran meliputi akurasi, presisi, *recall*, dan *F1-score* sebagai metrik evaluasi kinerja model. Akurasi mewakili proporsi prediksi yang benar secara keseluruhan. Dalam deteksi RD, akurasi tinggi menunjukkan bahwa model dapat mengklasifikasikan citra dengan benar sebagai RD atau non-RD. Namun, akurasi tinggi tidak selalu berarti model memiliki performa baik, terutama jika data tidak seimbang. Presisi mengukur proporsi prediksi positif yang benar. Dalam deteksi RD, presisi tinggi berarti bahwa model jarang menghasilkan hasil positif palsu, yang penting untuk menghindari diagnosis yang salah. *Recall* mengukur proporsi kasus RD yang benar-benar terdeteksi. Dalam deteksi RD, *recall* tinggi berarti model tidak melewatkan banyak kasus RD. *F1-Score* merupakan metrik yang menggabungkan presisi dan *recall*. *F1-Score* tinggi menunjukkan keseimbangan antara presisi dan *recall* yang baik.

Pengukuran ini dilakukan untuk memahami seberapa baik kemampuan model dalam mengklasifikasikan data, baik dalam mengidentifikasi kelas positif maupun negatif. Hasil pengujian ini memberikan wawasan yang komprehensif tentang performa relatif dari masing-masing arsitektur model. Dengan membandingkan metrik-metrik evaluasi tersebut, diperoleh hasil evaluasi keandalan dan efektivitas masing-masing model dalam menangani tugas klasifikasi. Berikut hasil evaluasi dari keempat model yang digunakan.

Tabel 3. Hasil evaluasi pengujian tiap model

Model	Akurasi (%)	Presisi (%)	*Recall (%)*	*F1-Score (%)*
ResNet152V2	93	94	94	93
Xception	96	96	96	96
DenseNet201	96	96	96	96
InceptionV3	94	94	94	94

Berdasarkan hasil evaluasi pengujian yang dilakukan terhadap empat model berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3* pada, dapat disimpulkan bahwa performa keempat model tersebut relatif

tinggi. *Xception* dan *DenseNet201* menunjukkan performa tertinggi dengan tingkat akurasi, presisi, *recall*, dan *F1 Score* sebesar 96%. Hal ini menunjukkan bahwa kedua model tersebut mampu dengan baik mengklasifikasikan data, baik dalam mengidentifikasi kelas positif maupun negatif, dengan sedikit kesalahan klasifikasi. Meskipun demikian, *ResNet152V2* dan *InceptionV3* juga menunjukkan performa yang baik dengan nilai akurasi, presisi, *recall*, dan *F1-Score* sebesar 93% dan 94% secara berturut-turut. Meskipun terdapat sedikit perbedaan dalam metrik evaluasi, namun hasil ini menegaskan bahwa keempat model tersebut dapat diandalkan untuk tugas klasifikasi. Hasil dari pengujian yang didapat hampir sama penelitian yang dilakukan oleh [9] dengan menggunakan *CNN transfer learning* untuk mendeteksi citra retina dalam konteks diagnosis RD yang menunjukan hasil *ResNet50* mendapat akurasi sebesar 90% pada dataset *APTOS 2019 BD*, sedangkan *VGG16* dan *InceptionV3* masing-masing mendapat akurasi sebesar 90% dan 89% pada dataset *E-Ophtha,* sehingga data ini menunjukan bahwa hasil pengujian yang terbaru dengan menggunakan empat model berbeda, yaitu *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3* memiliki hasil akurasi yang lebih tinggi di semua performa.

IV. KESIMPULAN

Keberhasilan keempat model yang telah dievaluasi dalam mengklasifikasikan penyakit RD dapat dilihat pada hasil pengujian yang diperoleh. Melalui pengujian yang teliti, *ResNet152V2*, *Xception*, *DenseNet201*, dan *InceptionV3* menunjukkan kemampuan yang mengesankan dalam membedakan gambar mata yang sehat dan yang terkena penyakit. Model-model ini mampu memberikan kontribusi signifikan dalam deteksi dini dan manajemen penyakit ini, yang merupakan langkah penting untuk mencegah komplikasi serius yang mungkin terjadi. Dari keempat model tersebut, *Xception* dan *DenseNet201* menonjol sebagai model terbaik dengan akurasi, presisi, *recall*, dan *F1-Score* mencapai 96%. Hal ini menandakan bahwa kedua model ini dapat diandalkan dalam memberikan diagnosis yang akurat, memberikan manfaat yang besar bagi dunia kesehatan terutama penyedia layanan kesehatan dan penderita penyakit RD.

Keberhasilan ini menunjukkan potensi besar dari pendekatan berbasis *deep learning* dalam mendukung diagnosis medis, membantu tenaga kesehatan dalam pengambilan keputusan yang lebih baik, dan meningkatkan perawatan pasien secara keseluruhan. Tentunya penelitian ini akan membuka potensi untuk penelitian selajutnya dalam mengembangkan model yang lebih baik, seperti dengan menggunakan dataset yang lebih lengkap dan menggunakan metode yang lebih baik untuk meningkatkan dan memperluas pengetahuan dalam bidang deteksi RD menggunakan *deep learning*. Dengan demikian, hasil evaluasi ini memberikan pandangan optimis tentang peran teknologi dalam perbaikan layanan kesehatan, khususnya dalam diagnosis penyakit mata yang kompleks seperti RD.

DAFTAR PUSTAKA

[1] K. Boyd, "Diabetic Retinopathy: Causes, Symptoms, Treatment," *American Academy of Ophthalmology*. 2023. [Online]. Available: https://www.aao.org/eye-health/diseases/what-is-diabetic-retinopathy
[2] P. A. W. Sebastian, Made Agus Kusumadjaja, I. G. A. M. Juliari, and I. G. A. R. Suryaningrum, "Karakteristik pasien diabetic retinopathy dengan dislipidemia di RSUP Sanglah Denpasar," *Intisari Sains Medis*, vol. 14, no. 1, pp. 59–63, 2023, doi: 10.15562/ism.v14i1.1576.
[3] A. R. Bhavsar, G. G. Emerson, M. V. Emerson, and D. J. Browning, "Epidemiology of diabetic retinopathy," *Diabet. Retin. Evidence-Based Manag.*, pp. 53–75, 2010, doi: 10.1007/978-0-387-85900-2_3.
[4] M. Tilahun, T. Gobena, D. Dereje, M. Welde, and G. Yideg, "Prevalence of diabetic retinopathy and its associated factors among diabetic patients at debre markos referral hospital, Northwest Ethiopia, 2019: Hospital-based cross-sectional study," *Diabetes, Metab. Syndr. Obes.*, vol. 13, pp. 2179–2187, 2020, doi: 10.2147/DMSO.S260694.
[5] Kemenkes RI, "PEDOMAN NASIONAL PELAYANAN KEDOKTERAN TATA LAKSANA RETINOPATI DIABETIKA," 2023.
[6] N. Han, J. Yao, and S. Li, "Accelerated Life Test and Life Prediction of an Electromechanical Actuator," *Proc. - 2019 Int. Conf. Artif. Intell. Adv. Manuf. AIAM 2019*, pp. 647–651, 2019, doi: 10.1109/AIAM48774.2019.00134.
[7] R. Yasashvini, V. Raja Sarobin M, R. Panjanathan, S. Graceline Jasmine, and L. Jani Anbarasi, "Diabetic Retinopathy Classification Using CNN and Hybrid Deep Convolutional Neural Networks," *Symmetry (Basel).*, vol. 14, no. 9, 2022, doi: 10.3390/sym14091932.
[8] D. Le *et al.*, "Transfer Learning for Automated OCTA Detection of Diabetic Retinopathy," *Transl. Vis. Sci. Technol.*, vol. 9, no. 2, pp. 35–35, Jan. 2020, doi: 10.1167/TVST.9.2.35.
[9] Y. S. Devi and S. P. Kumar, "A deep transfer learning approach for identification of diabetic retinopathy using data augmentation," *IAES Int. J. Artif. Intell. (IJ-AI*, vol. 11, no. 4, pp. 1287–1296, 2022, doi: 10.11591/ijai.v11.i4.pp1287-1296.
[10] Y. Yu, H. Lin, J. Meng, X. Wei, H. Guo, and Z. Zhao, "Deep transfer learning for modality classification of medical images," *Inf.*, vol. 8, no. 3, 2017, doi: 10.3390/info8030091.
[11] S. Kulshrestha, "What Is A Convolutional Neural Network?," *Developing an Image Classifier Using TensorFlow*. 2019. doi: 10.1007/978-1-4842-5572-8_6.
[12] W. Supriyanti and D. A. Anggoro, "Classification of Pandavas Figure in Shadow Puppet Images using Convolutional Neural Networks," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 7, no. 1, 2021, doi: 10.23917/khif.v7i1.12484.
[13] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.
[14] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, pp. 1–6, 2017, doi: 10.1109/ICEngTechnol.2017.8308186.
[15] K. Zakka, "CS231n Convolutional Neural Networks for Visual Recognition," *Stanford Univ.*, pp. 1–18, 2021, [Online]. Available: https://cs231n.github.io/convolutional-network/#add
[16] W. Setiawan, M. I. Utoyo, and R. Rulaningtyas, "Transfer learning with multiple pre-trained network for fundus classification," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, pp. 1382–1388, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14868.

[17] KOUSTUBH, "ResNet , AlexNet , VGGNet , Inception : Understanding various architectures of Convolutional Networks." pp. 1–7, 2018. [Online]. Available: https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/
[18] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," *ISCAS 2010 - 2010 IEEE Int. Symp. Circuits Syst. Nano-Bio Circuit Fabr. Syst.*, pp. 253–256, 2010, doi: 10.1109/ISCAS.2010.5537907.
[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.
[20] M. F. Naufal and S. F. Kusuma, "Pendeteksi Citra Masker Wajah Menggunakan CNN dan Transfer Learning," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 6, p. 1293, 2021, doi: 10.25126/jtiik.2021865201.
[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.
[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.
[23] Asia Pacific Tele-Ophthalmology Society, "APTOS 2019 blindness detection," *Kaggle Competition*. 2019. [Online]. Available: https://www.kaggle.com/competitions/aptos2019-blindness-detection
[24] N. Sikder, M. S. Chowdhury, A. Shamim Mohammad Arif, and A. Al Nahid, "Early blindness detection based on retinal images using ensemble learning," *2019 22nd Int. Conf. Comput. Inf. Technol. ICCIT 2019*, 2019, doi: 10.1109/ICCIT48885.2019.9038439.
[25] R. Rismiyati and A. Luthfiarta, "VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification," *Telematika*, vol. 18, no. 1, p. 37, 2021, doi: 10.31315/telematika.v18i1.4025.
[26] N. E. M. Khalifa, M. Loey, and M. H. N. Taha, "Insect pests recognition based on deep transfer learning models," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 1, pp. 60–68, 2020.