

IMPLEMENTASI *ENSEMBLE WEIGHTED VOTING* PADA ARSITEKTUR *DENSENET MOBILENET XCEPTION* UNTUK KLASIFIKASI PENYAKIT *DIABETIC RETINOPATHY*

**Lucky Indra Kesuma^{1*}, Des Alwine Zayanti², Anita Desiani³, Purwita Sari⁴, Zulhipni Reno Saputra⁵,
Muhammad Ihsan⁶, Fathona Nur Muzayyadah⁷**

^{1,5,6} Teknologi Informasi, Fakultas Teknik, Universitas Muhammadiyah Palembang, Palembang, Indonesia

^{2,3,7} Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sriwijaya, Palembang, Indonesia

⁴ Manajemen Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya, Palembang, Indonesia

Email: ^{1*}lucky@um-palembang.ac.id, ²desalwinez@unsri.ac.id, ³anita_desiani@unsri.ac.id, ⁴wita@ilkom.unsri.ac.id,

⁵zulhipni_renosaputra@um-palembang.ac.id, ⁶ihsan_idris@um-palembang.ac.id, ⁷fathonanur.muza7@gmail.com

(* : corresponding author)

Abstrak—*Convolutional Neural Network (CNN)* merupakan salah satu pendekatan *deep learning* yang banyak digunakan pada tugas klasifikasi dan segmentasi citra, termasuk pada bidang kesehatan. Salah satu penerapan penting CNN adalah pada analisis citra *Diabetic Retinopathy (DR)*, yaitu penyakit pada retina mata yang disebabkan oleh komplikasi diabetes jangka panjang dan dapat menyebabkan gangguan penglihatan hingga kebutaan apabila tidak terdeteksi secara dini. Namun, penggunaan arsitektur CNN tunggal sering mengalami keterbatasan, seperti *overfitting*, kebutuhan komputasi yang tinggi, atau kemampuan ekstraksi fitur yang belum optimal. Oleh karena itu, metode *ensemble* dapat digunakan untuk menggabungkan keunggulan dari beberapa model guna meningkatkan kinerja klasifikasi. Pada penelitian ini diusulkan metode *ensemble* berbasis *weighted voting* dengan menggabungkan tiga arsitektur CNN, yaitu *DenseNet*, *MobileNet*, dan *Xception*, untuk klasifikasi biner *Diabetic Retinopathy*. *DenseNet* dipilih karena kemampuannya dalam mengekstraksi fitur yang kaya melalui konektivitas antar lapisan, *MobileNet* dipilih karena efisiensi komputasi dan ukuran model yang ringan, sedangkan *Xception* digunakan karena kemampuannya menyeimbangkan kedalaman jaringan dan efisiensi komputasi melalui *depthwise separable convolution*. Tahapan penelitian meliputi pengumpulan data, pelatihan model, pengujian, serta evaluasi kinerja. Dataset *EyePACS* digunakan sebagai data pelatihan, sedangkan dataset *APTOS* dimanfaatkan sebagai data pengujian untuk menguji kemampuan generalisasi model. Hasil eksperimen menunjukkan bahwa metode *ensemble* yang diusulkan menghasilkan kinerja yang baik dengan nilai akurasi sebesar 85.22%, sensitivitas 70.63%, spesifisitas 99.40%, *F1-score* 87.21%, serta nilai *Cohen's Kappa* sebesar 0.7032. Hasil ini menunjukkan bahwa pendekatan *ensemble* mampu meningkatkan kinerja klasifikasi dan mengurangi permasalahan *overfitting* dibandingkan model CNN tunggal, serta berpotensi dikembangkan sebagai sistem pendukung keputusan untuk *skrining otomatis Diabetic Retinopathy*.

Kata Kunci: *DenseNet, Ensemble, Diabetic Retinopathy, MobileNet, Xception*

Abstract—*Convolutional Neural Networks (CNNs)* have been widely applied in image-based classification and segmentation tasks, including medical image analysis. One important application is the classification of *Diabetic Retinopathy (DR)*, a retinal disease caused by long-term complications of diabetes that can lead to visual impairment or blindness if not detected early. However, single CNN architecture often suffers from limitations such as *overfitting*, high computational cost, or insufficient feature representation. To address these challenges, ensemble learning can be employed to combine the strengths of multiple CNN models and improve classification performance. In this study, an ensemble method based on *weighted voting* is proposed by integrating three CNN architectures, namely *DenseNet*, *MobileNet*, and *Xception*, for binary classification of *Diabetic Retinopathy*. *DenseNet* is selected for its ability to extract rich and discriminative features through dense layer connectivity, *MobileNet* is chosen for its computational efficiency and lightweight architecture, and *Xception* is employed for its balance between network depth and efficiency using *depthwise separable convolutions*. The research workflow includes data collection, model training, testing, and performance evaluation. The *EyePACS* dataset is used for training, while the *APTOS* dataset is utilized for testing to evaluate generalization capability. Experimental results demonstrate that the proposed ensemble method achieves an accuracy of 85.22%, sensitivity of 70.63%, specificity of 99.40%, an *F1-score* of 87.21%, and a *Cohen's Kappa* value of 0.7032. These results indicate that the ensemble approach effectively improves classification performance and reduces *overfitting* compared to single CNN models. The proposed method shows potential to be developed as a decision support system for automated *Diabetic Retinopathy* screening.

Keywords: *Diabetic Retinopathy, Ensemble, DenseNet, MobileNet, Xception*

1. PENDAHULUAN

Convolutional Neural Network (CNN) merupakan metode *deep learning* yang sering digunakan baik dalam segmentasi maupun klasifikasi [1]. CNN memiliki kelebihan yaitu dapat menangani data input berbentuk citra berukuran $m \times n$ [2]. Penerapan CNN banyak digunakan pada data gambar, salah satunya adalah citra *Diabetic Retinopathy (DR)* [3]. DR merupakan penyakit yang menyerang retina mata akibat komplikasi penyakit diabetes yang berkepanjangan dan tidak diobati sehingga menyebabkan kebutaan [4]. Pemeriksaan citra DR umumnya dilakukan secara manual sehingga memakan waktu yang relatif lama dan dibutuhkan ketelitian yang tinggi. Berdasarkan hal tersebut para peneliti melakukan suatu sistem otomatis untuk mendeteksi DR agar dapat diidentifikasi secara cepat dan akurat [5]. Beberapa

penelitian menggunakan CNN pada klasifikasi DR diantaranya Pratt *et al.* [6] dan [7] memakai dataset Kaggle serta Doshi *et al.* [8] dan Ghosh *et al.* [9] memakai dataset EyePACS.

Penyakit *Diabetic Retinopathy* (DR) merupakan salah satu penyebab utama gangguan penglihatan dan kebutaan pada penderita diabetes. Proses *skrining* DR secara konvensional masih bergantung pada pemeriksaan manual oleh tenaga medis, sehingga membutuhkan waktu yang relatif lama dan tingkat ketelitian yang tinggi. Kondisi ini menjadi tantangan terutama pada fasilitas kesehatan dengan keterbatasan jumlah tenaga ahli dan tingginya volume pemeriksaan. Oleh karena itu, pengembangan sistem klasifikasi otomatis berbasis citra fundus retina menjadi penting untuk membantu proses *skrining* awal DR secara cepat dan akurat. Pemanfaatan metode *deep learning*, khususnya CNN, diharapkan dapat meningkatkan efisiensi, konsistensi, dan objektivitas diagnosis serta mendukung pengambilan keputusan klinis.

CNN berkembang dengan berbagai macam arsitektur diantaranya *MobileNet*, *DenseNet*, dan *Xception*. *MobileNet* merupakan salah satu arsitektur CNN yang memiliki ukuran model yang ringan dan parameter yang kecil. *MobileNet* menggunakan *depthwise separable convolution* yang memisahkan konvolusi biasa terdiri dari *depthwise* dan *pointwise* sehingga arsitektur ini memiliki jumlah parameter yang kecil [10]. Namun *MobileNet* rentan *overfitting* karena fitur yang ditangkap lebih sedikit, sehingga hasilnya kurang akurat [11], [12]. Gao *et al.* [12] menggunakan *MobileNet* pada klasifikasi DR memakai dataset Messidor mendapatkan akurasi dan AUC diatas 90%. Sae-Lim *et al.* [10] menggunakan *MobileNet* melakukan klasifikasi lesi kulit pada dataset *Human Againsts Machine* mendapatkan nilai akurasi, sensitivitas, spesifisitas dan *f1-score* diatas 80%. Souid *et al.* [13] melakukan klasifikasi penyakit paru-paru pada dataset *Chest X-Ray* mendapatkan nilai akurasi dan spesifisitas diatas 90% sedangkan sensitivitas dan F1-Score di bawah 60%.

Arsitektur CNN lainnya yang dapat menangkap fitur lebih baik adalah *Densely Connected Convolutional Network* (*DenseNet*). *DenseNet* merupakan arsitektur yang memiliki koneksi langsung dari setiap lapisan ke semua lapisan berikutnya, sehingga dapat meningkatkan penggunaan fitur yang akan dipakai untuk lapisan selanjutnya [3], [14], [15]. Sayangnya *DenseNet* dapat menurunkan efisiensi komputasi karena meningkatnya jumlah fitur disetiap lapisan dan parameternya yang besar [16], [17]. Penelitian [18] melakukan klasifikasi DR memakai dataset Kaggle menghasilkan nilai akurasi sebesar 97.20%. [19] melakukan klasifikasi DR memakai dataset APTOS menghasilkan nilai akurasi 96%. [20] melakukan klasifikasi COVID-19 memakai dataset dari Rumah Sakit Sao Paulo Brazil menghasilkan nilai akurasi sebesar 92%. Namun ketiga penelitian tersebut hanya mengukur kinerja akurasi saja.

Berbeda dengan *DenseNet*, arsitektur *Xception* merupakan metode CNN dapat meningkatkan efisiensi pada proses komputasi. *Xception* juga memiliki *depthwise separable convolution* seperti pada arsitektur *MobileNet*, tetapi arsitektur ini memiliki koneksi residual dalam arsitekturnya. Arsitektur *Xception* memiliki parameter yang kecil dan efisien secara komputasi [15], [21]. Kelemahan *Xception* yaitu membutuhkan data yang banyak untuk dapat melatih semua parameternya [15]. Kassani *et al.* [22] melakukan klasifikasi DR memakai dataset APTOS sebanyak 3662 data menghasilkan nilai akurasi, sensitivitas dan spesifisitas diatas 80%. Chollet [23] dalam mengklasifikasikan DR memakai dataset JFT sebanyak 16000 mendapatkan akurasi sebesar 94.5% namun tidak mempertimbangkan ukuran kinerja yang lain.

Beberapa penelitian sebelumnya telah menerapkan arsitektur CNN tunggal maupun metode *ensemble* untuk klasifikasi *Diabetic Retinopathy*. Namun, sebagian besar penelitian tersebut masih berfokus pada peningkatan nilai akurasi tanpa mempertimbangkan metrik evaluasi lain yang penting, seperti sensitivitas, spesifisitas, *F1-score*, dan Cohen's Kappa, terutama pada kondisi data yang tidak seimbang. Selain itu, beberapa pendekatan *ensemble* menggunakan arsitektur CNN yang kompleks sehingga meningkatkan beban komputasi dan berpotensi menimbulkan *overfitting*. Berbeda dengan penelitian sebelumnya, penelitian ini mengusulkan metode *ensemble* berbasis *weighted voting* dengan mengombinasikan tiga arsitektur CNN yang memiliki karakteristik saling melengkapi, yaitu *DenseNet*, *MobileNet*, dan *Xception*, dengan evaluasi kinerja yang lebih komprehensif.

Setiap arsitektur memiliki kelemahan dan keunggulan masing-masing, salah satu metode yang dapat mengambil keunggulan dari arsitektur yaitu metode *ensemble*. Metode *ensemble* menggabungkan hasil kinerja dari beberapa metode klasifikasi tunggal untuk meningkatkan nilai kinerja pada klasifikasi [24]. Adapun beberapa penelitian yang menggunakan metode *ensemble* dalam mengklasifikasikan DR diantaranya Jiang *et al.* [25] menerapkan metode *ensemble* menggunakan arsitektur *InceptionV3*, *InceptionResNetV2*, dan *ResNet152* dengan akurasi yang naik sebesar 1,11% dari rata-rata akurasi sebelumnya menjadi 88.21%. Jinfeng *et al.* [26] menerapkan metode *ensemble* pada arsitektur *DenseNet121*, *ResNet50*, dan *InceptionV3* dengan akurasi yang naik sebesar 5,37% dari rata-rata akurasi sebelumnya menjadi 80,36%. Nguyen *et al.* [8] menerapkan metode *ensemble* dengan arsitektur VGG-16 dan VGG-19 dengan akurasi, sensitivitas dan spesifisitas diatas 80%. Hagos and Kant [27] menerapkan metode *ensemble* pada arsitektur *Inception V3* dan QIY model menghasilkan akurasi 90%.

Metode *ensemble* memiliki beberapa teknik dalam pengambilan keputusan prediksi yaitu *bagging*, *stacking*, *boosting*, dan *voting* [28]. Teknik pengambilan keputusan yang paling sederhana adalah *voting* [29]. Salah satu teknik *voting* yang sering dipakai adalah *weighted voting* [24]. Teknik ini menggunakan bobot hasil dari tiap arsitektur untuk menentukan hasil prediksi label.

Berdasarkan kelebihan dari masing-masing arsitektur, maka pada Penelitian ini akan menerapkan metode *ensemble* pada hasil kinerja dari proses latih dari arsitektur *DenseNet*, *MobileNet* dan *Xception* dengan pengambilan

keputusan menggunakan teknik *weighted voting*. Evaluasi hasil kinerja dilihat berdasarkan nilai akurasi, sensitivitas, spesifisitas, *f1-score*, dan *cohens kappa*.

2. METODE PENELITIAN

2.1. Dataset EyePACS dan APTOS

Data yang digunakan pada penelitian ini menggunakan 2 dataset, yang pertama adalah dataset EyePACS yang diperoleh dari laman <https://www.kaggle.com/c/diabetic-retinopathy-detection> terdiri dari 35126 data retina. Pada dataset EyePACS dibagi menjadi 2 label, yaitu 25810 mata normal dan 9316 data mata yang mengalami DR. Dataset APTOS diperoleh dari laman <https://www.kaggle.com/c/blindness-detection/data> yang terdiri dari 3662 data retina. Pada dataset APTOS dibagi menjadi 2 label yaitu, 1805 mata normal dan 1857 DR. Dataset EyePACS memiliki data yang tidak seimbang diantara label normal dan label DR, sehingga pada penelitian ini diambil 9000 data untuk mata normal dan 9000 untuk data mata yang mengalami DR agar seimbang.

2.2. Densely Connected Convolutional Network (DenseNet)

Densely Connected Convolutional Network (DenseNet) memiliki konektivitas yang padat dan lebih unggul, yang artinya koneksi langsung dari setiap lapisan ke setiap lapisan berikutnya, sehingga dapat meningkatkan aliran informasi antar lapisan. Arsitektur *DenseNet* merancang *dense block* yang dapat digunakan untuk operasi penggabungan dengan *concatenate* agar fitur dari lapisan sebelumnya dapat digunakan kembali [16], [30]. Lapisan antara dua *dense blocks* adalah lapisan transisi yang terdiri dari *convolutional layer* dan *pooling layer*. Arsitektur ini menggabungkan hasil dari fitur layer sebelumnya dengan fitur *input*. Oleh karena itu *DenseNet* dapat meningkatkan penggunaan fitur dan mengurangi parameter dengan sangat baik [14], [18], [31].

2.3. MobileNet

MobileNet merupakan arsitektur yang menerapkan *depthwise separable convolution*. *MobileNet* menggunakan *depthwise separable convolution* untuk menerapkan *depthwise* pada setiap saluran *input*, dan kemudian digunakan konvolusi 1×1 (*pointwise*) untuk membuat kombinasi dari *output depthwise convolution* [11]. Berdasarkan hal tersebut, komputasi pada *MobileNet* lebih efisien. *MobileNet* dapat menunjukkan kekuatan kinerja meskipun arsitektur ini menggunakan jumlah parameter yang lebih kecil.

2.4. Xception

Xception merupakan singkatan dari *extreme inception* yang dibuat oleh Francois Chollet [23]. *Xception* menggunakan *depthwise separable convolution* pada arsitekturnya namun memiliki koneksi residual, sehingga arsitektur ini memiliki parameter yang kecil. *Xception* memiliki 36 lapisan konvolusi untuk mengekstrak fitur [23]. *Xception* memiliki 36 lapisan konvolusi yang disusun menjadi 14 modul dan memiliki koneksi residual linier, kecuali untuk modul pertama dan terakhir [22]. Input masuk melalui *entry flow*, kemudian melalui *middle flow* yang diulang 8 kali, dan masuk ke *exit flow*. Semua lapisan konvolusi dan *separable convolution* diikuti oleh *batch normalization*. *Xception* memperkenalkan revolusi *depthwise separable* yang dapat meningkatkan kinerja model tanpa meningkatkan kompleksitas model.

2.5. Proses Pelatihan Data

Pada tahap ini digunakan 18000 data latih berukuran 336×336 . Dari data tersebut akan dibagi menjadi 17280 data latih dan 720 data validasi. Pada proses pelatihan akan dilakukan untuk melatih data dengan tujuan mendapatkan bobot terbaik. Masing-masing arsitektur yang digunakan adalah *DenseNet*, *MobileNet* dan *Xception* pada proses pelatihan. Hasil dari tahap pelatihan akan disimpan bobot terbaik pada masing-masing arsitektur. Pada tahapan ini digunakan jumlah *epoch* sebanyak 30, sehingga iterasi atau jumlah *batch* dalam satu *epoch* sebanyak 2468.

Pada arsitektur yang diusulkan tersebut terdapat beberapa proses yang yaitu *convolution layer*, *batch normalization*, fungsi aktivasi *ReLU*, *max pooling*, *concatenate*, fungsi aktivasi *sigmoid*, fungsi aktivasi *softmax*, dan *depthwise separable convolution* dan *loss function binary cross entropy*.

1. Convolution Layer

Convolutional layer merupakan blok utama yang terdapat pada CNN dengan tujuan untuk mempelajari fitur dari *input*, seperti gambar digital. Pada *convolutional layer* terdapat *stride*, dan ukuran *stride* biasanya ditentukan oleh pembuat model. Untuk menghitung hasil dari *convolution layer* dapat menggunakan Persamaan (1) [32]:

$$a_{ij} = \left(\sum_{u=0}^{n-1} \sum_{v=0}^{n-1} (c_{u+i,v+j} \times k_{u+1,v+1}) \right) + b_q \quad (1)$$

dimana i merupakan baris, j adalah kolom, n adalah ukuran tinggi kernel, $*$ merupakan operasi konvolusi, C_p merupakan matriks *input* ke- p , K_q adalah matriks kernel ke- q , b_q adalah bias untuk kernel ke- q , A_q^p merupakan matriks hasil operasi konvolusi untuk kernel ke- q pada *input* ke- p , dan D_q^p merupakan matriks hasil konvolusi (*feature maps*) kernel ke- q pada *input* ke- p .

2. Batch Normalization

Batch normalization dapat mempercepat pelatihan suatu jaringan dan dapat meningkatkan ketahanan terhadap *overfitting* [33]. *Batch normalization* bekerja dengan menormalkan *input* dari setiap lapisan yang dipakai [15]. Untuk menghitung hasil *batch normalization* dapat menggunakan Persamaan (2), (3), dan (4) [34].

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij} \quad (2)$$

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2 \quad (3)$$

$$x_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}} \quad (4)$$

Untuk $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, dimana n adalah jumlah *mini-batch* (kolom), m adalah banyaknya data dalam satu *mini-batch* (baris), dan x_{ij} merupakan entri matriks *input* pada baris ke- i dan kolom ke- j , $\hat{x}_{ij}$ merupakan entri matriks *input* yang telah dinormalisasi pada baris ke- i dan kolom ke- j , serta ε merupakan nilai konstanta positif terkecil.

3. Fungsi Aktivasi

Fungsi Aktivasi termasuk dari lapisan *non linier* yang biasanya mengikuti *convolution layer*. Fungsi aktivasi merupakan bagian yang sangat penting karena memungkinkan jaringan agar lebih kuat terhadap suatu fungsi yang kompleks [15]. Fungsi aktivasi memiliki beberapa jenis diantaranya *Rectified Linear Unit (ReLU)*, *sigmoid*, dan *softmax*. Secara matematis, *ReLU* dapat didefinisikan pada Persamaan 5 [15]:

$$f(x) = \max(0, x) \quad (5)$$

ReLU merupakan salah satu fungsi aktivasi yang digunakan pada CNN yang apabila *input* fungsi aktivasi bernilai negatif maka *output* berubah menjadi nol. Sedangkan apabila *input* fungsi aktivasi bernilai positif, maka *output* adalah nilai *input* itu sendiri.

Fungsi aktivasi lainnya adalah *sigmoid*. *Input* dari fungsi aktivasi *sigmoid* adalah bilangan *real* dan *output*nya adalah bilangan yang terletak antara 0 sampai 1. Secara matematis *sigmoid* didefinisikan sebagai Persamaan (6), dimana z adalah *input sigmoid* dan $s(z) \in (0, 1)$.

$$s(z) = \frac{1}{1 + e^{-z}} \quad (6)$$

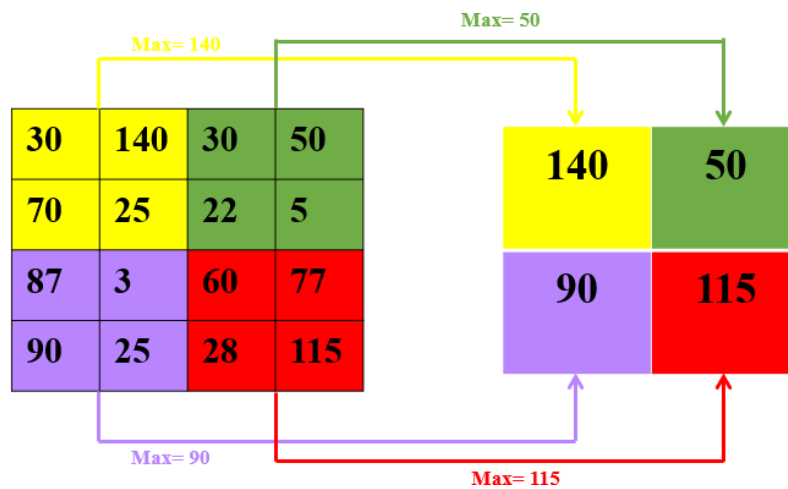
Fungsi aktivasi yang digunakan dalam lapisan akhir (*output*) adalah fungsi aktivasi *softmax*. Fungsi aktivasi *softmax* biasanya digunakan untuk menghitung probabilitas dalam klasifikasi *multiclass*. Nilai fungsi aktivasi *softmax* terletak antara nilai 0 dan 1. Fungsi aktivasi *softmax* dapat dilihat pada Persamaan (7) [35]:

$$p(x_i) = \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}} \quad (7)$$

Dimana $p(x_i)$ adalah probabilitas atau nilai fungsi. x_i adalah *output* dari bobot yang diberikan sistem dan $i = 1, 2, \dots, n$ dan $k = 1, 2, \dots, n$.

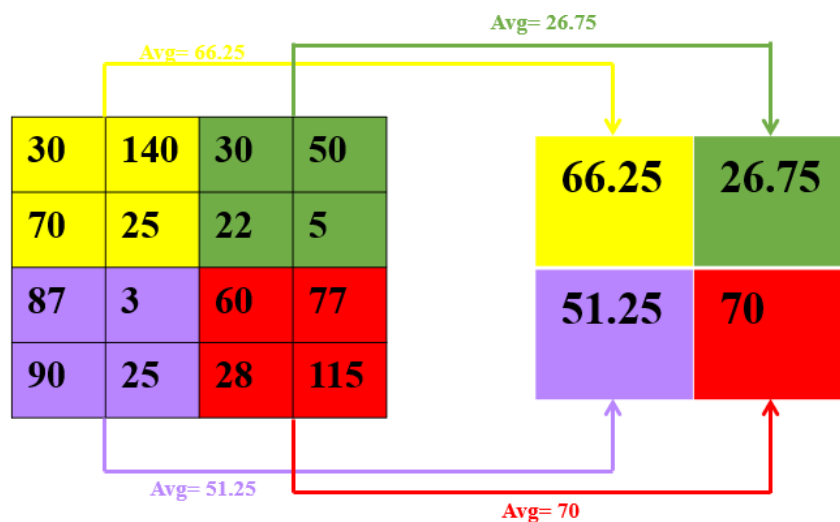
4. Pooling Layer

Pooling layer bertujuan untuk menurunkan beban komputasi dengan cara mengurangi ukuran *input* data yang biasanya ditempatkan setelah *convolutional layer* [36]. Ada banyak jenis *pooling layer* yang digunakan seperti *max pooling*, *average pooling*, dan *global average pooling*. *Max pooling* mengambil nilai maksimum pada setiap *grid* yang telah dipilih [15]. Ilustrasi *max pooling* dapat dilihat pada Gambar 1.



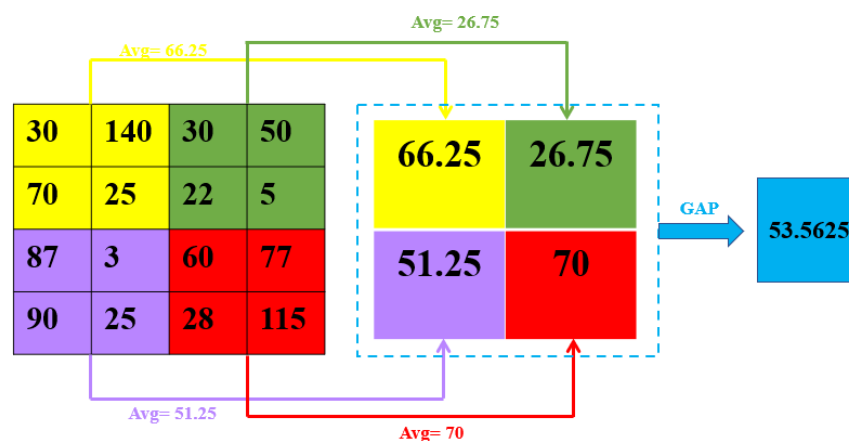
Gambar 1. Ilustrasi Max pooling

Average pooling layer bekerja dengan cara mengambil nilai rata-rata pada setiap grid yang telah dipilih [15]. Ilustrasi average pooling dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi average pooling

Adapun global average pooling digunakan untuk mencegah *overfitting* pada model dan mengurangi parameter pada model. Global average pooling bekerja dengan mengambil nilai rata-rata pada setiap grid kemudian merubah ukuran menjadi 1×1 . Ilustrasi global average pooling dapat dilihat pada Gambar 3.



Gambar 3. Ilustrasi Global Average Pooling

5. Concatenate Layer

Concatenate layer merupakan gabungan lapisan yang digunakan untuk mengkoneksikan informasi antar lapisan [37]. Pada arsitektur *DenseNet*, *concatenate* digunakan sebagai penggunaan peta fitur dari lapisan sebelumnya sebagai input [30].

6. Loss Function: Binary Cross Entropy

Loss function dipilih untuk melatih model namun harus diminimalkan agar dapat mencapai akurasi yang lebih baik dan dapat mengukur nilai kerugian yang diperoleh dalam proses pelatihan [36]. *Binary cross entropy* adalah *loss function* yang banyak digunakan untuk klasifikasi biner [38]. *Loss function binary cross entropy* dapat dihitung menggunakan Persamaan (8) [39]:

$$L = \frac{-1}{n} \sum_{i=0}^n ((y_i \log(d_i)) + (1 - y_i) \log(1 - (d_i))) \quad (8)$$

dimana n menunjukkan banyaknya arsitektur. y_i adalah label kelas sebenarnya dimana bernilai 1 untuk label DR dan 0 untuk normal. d_i adalah nilai prediksi. L adalah *loss function binary cross entropy*.

7. Depthwise Separable Convolution

Depthwise separable convolution merupakan layer berbentuk blok yang terdiri dari *depthwise convolution* dan *pointwise convolution* (konvolusi 1×1). *Depthwise separable convolution* digunakan untuk mengurangi jumlah parameter dan komputasi yang digunakan dalam operasi konvolusi [40]. *Depthwise convolution* dapat dihitung menggunakan Persamaan (9) [31]:

$$A_{i(i,j)} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} (W_{i(u+i,v+j)} \times k_{i(u+1,v+1)}) \quad (9)$$

Dimana A_i adalah *depthwise convolution*, k adalah kernel, W adalah matriks input dengan $i = 1, 2, \dots, n$ dan $j = 1, 2, \dots, n$. *Pointwise convolution* dapat dihitung menggunakan Persamaan (10) [31]:

$$A_B = \sum_{u=0}^{n-1} (t_{i(i,j,u+1)}) \times (k_{i(i,j,u+1)}) \quad (10)$$

Dimana B adalah *pointwise convolution*, k adalah kernel, t adalah input matriks dengan $i = 1, 2, \dots, n$ dan $j = 1, 2, \dots, n$.

8. Weighted Voting

Teknik *weighted voting* merupakan teknik untuk menentukan hasil prediksi menggunakan bobot. Cara kerja *weighted voting* adalah sistem memberi bobot untuk output pada masing-masing arsitektur yang digunakan, nilai bobot dimasukkan dalam fungsi aktivasi *softmax*. Kemudian hasilnya di kali dengan bobot pada masing-masing arsitektur sesuai dengan kinerja masing-masing arsitektur. Lalu hasil dari seluruh bobot output dijumlahkan. Untuk menentukan hasil prediksi menggunakan teknik *weighted voting* dapat dihitung menggunakan Persamaan (11) [41].

$$F(x_i) = \sum_{i=1}^n w_i p(x_i) \quad (11)$$

Dimana $F(x_i)$ adalah hasil prediksi kelas, w_i adalah bobot hasil proses pelatihan dengan $i = 1, 2, \dots, n$, dan $p(x_i)$ adalah fungsi *softmax*, untuk $F(x_i) \in [0, 1]$.

2.6. Proses Pengujian Data

Pada tahap ini bobot dari masing-masing arsitektur diimplementasikan kedalam *ensemble*. Hasil dari proses pengujian dari tiap arsitektur akan disimpan dan dibandingkan dengan *ensemble*. Hasil prediksi berupa bobot dengan rentang nilai $[0, 1]$. Hasil prediksi dikonversi dimana input bobot bernilai $0 \leq x < 0,5$ maka output bobot adalah 0 yang berarti normal, sedangkan input bobot bernilai $0,5 \leq x \leq 1$ maka output bobot adalah 1 yang berarti DR [42],[43],[44]. Probabilitas tersebut akan masuk ke dalam *weighted voting layer*, dimana ketiga nilai probabilitas akan diolah menggunakan bobot dalam layer dan fungsi *softmax*. Model *ensemble* akan mengeluarkan output probabilitas yang baru, dimana hasil output akan diolah dengan *loss function binary cross entropy* dan bobot pada *weighted voting layer* akan diperbaiki sampai mendapatkan bobot terbaik.

2.7. Evaluasi

Terdapat beberapa evaluasi kinerja untuk mengetahui performa dari metode yang diusulkan yaitu akurasi, sensitivitas, spesifisitas, *F1-Score*, dan *cohens kappa*. Akurasi adalah ukuran yang menunjukkan seberapa baik model

klasifikasi dengan benar mengidentifikasi objek. Sensitivitas adalah ukuran proporsi TP yang diidentifikasi dengan benar. Spesifisitas adalah ukuran proporsi TN yang diidentifikasi dengan benar sebagai negatif [45]. *F1-score* adalah rata-rata dari nilai sensitivitas dan spesifisitas. *Cohens kappa* adalah ukuran untuk melihat tingkat kesesuaian antara hasil prediksi yang dihasilkan dengan label sebenarnya pada skala nominal dengan dua kelas atau lebih [46]. Untuk menghitung nilai akurasi, sensitivitas dan spesifisitas, *f1-score* dan *cohens kappa* seperti pada persamaan (12), (13), (14), (15) [47] dan *cohens kappa* pada persamaan (16) [46].

$$Akurasi = \frac{TP + TN}{TP + FN + TN + FP} \times 100\% \quad (12)$$

$$Sensitivitas = \frac{TP}{TP + FN} \times 100\% \quad (13)$$

$$Spesifisitas = \frac{TN}{TN + FP} \times 100\% \quad (14)$$

$$F1Score = \frac{2TP}{2TP + FP + FN} \times 100\% \quad (15)$$

$$x = \frac{2((TP \times TN) - (FP \times FN))}{((TP + FP) \times (FP + TN)) + ((TP + FN) \times (FN + TN))} \quad (16)$$

Nilai akurasi, sensitivitas, spesifisitas, *F1-Score*, dan *IoU* dikategorikan dalam beberapa kategori yang dapat dilihat pada Tabel 1.

Tabel 1. Kategori nilai kinerja arsitektur [51]

Nilai Kinerja Model (NK) (%)	Kategori
91%-100%	Sangat Baik
81%-90%	Baik
71%-80%	Cukup
61%-70%	Kurang baik
≤60%	Gagal

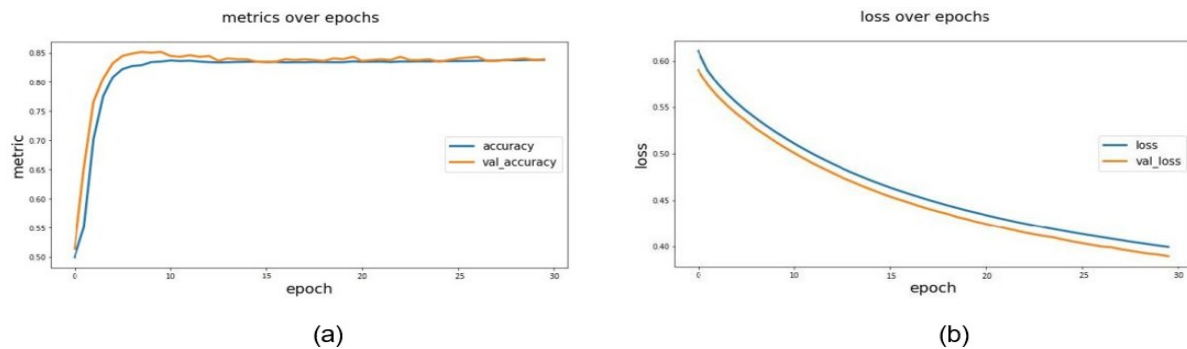
Berdasarkan Tabel 1 dapat dilihat kategori nilai kinerja arsitektur dibagi menjadi 5 yaitu gagal, kurang baik, cukup baik, baik, dan sangat baik.

3. HASIL DAN PEMBAHASAN

3.1. Tahap Pelatihan

Pada tahap ini digunakan 18000 data latih berukuran 336×336 . Dari data tersebut akan dibagi menjadi 17280 data latih dan 720 data validasi. Tahap ini menggunakan jumlah *epoch* sebanyak 30 dan jumlah *batch* dalam 1 *epoch* sebanyak 540. Proses pelatihan pada model *ensemble* akan dilakukan sampai *epoch* ke-30 untuk mendapatkan bobot terbaik yang akan digunakan pada proses pengujian. Pada model arsitektur CNN, akan dilakukan inisialisasi nilai bobot pada setiap *epoch* kemudian menghitung nilai *loss* untuk setiap *epoch*. Bobot disimpan apabila nilai *loss* pada data validasi lebih kecil dari *epoch* sebelumnya, jika tidak maka bobot akan diperbarui untuk *epoch* selanjutnya sampai menghasilkan bobot terbaik. Grafik hasil proses pelatihan data uji dan data validasi untuk nilai akurasi dan nilai *loss* dapat dilihat pada Gambar 4(a) dan 4(b).

Hasil akurasi dan *loss* dari proses pelatihan menggunakan model yang diusulkan dapat dilihat pada Gambar 4 Hasil akurasi pada data latih dan validasi ditunjukkan pada Gambar 4(a). Pada Gambar 4(a) grafik akurasi model *ensemble* untuk data latih selalu meningkat disetiap *epoch*. *Epoch* pertama akurasi diperoleh sebesar 0,4996 dan akan meningkat terus menerus menuju 0,8377. Nilai akurasi untuk data validasi mengalami kenaikan terus menerus hingga *epoch* ke-10 kemudian mengalami penurunan pada *epoch* ke-11 dan nilai akurasinya stabil sampai *epoch* ke-30. Berdasarkan grafik pada Gambar 4(a) model *Ensemble* tidak mengalami *overfitting* karena nilai akurasi untuk data latih dan untuk data validasi sama-sama meningkat. Pada Gambar 4(b) Grafik *loss* arsitektur *ensemble* untuk data latih dan data validasi sama-sama mengalami penurunan pada setiap *epoch* hingga *epoch* ke-50. Nilai akurasi dan *loss* yang didapat pada proses pelatihan, dapat dikatakan baik karena akurasi model sudah lebih dari 81% dan nilai *loss* mendekati 0. Berdasarkan grafik pada Gambar 4(b) model *Ensemble* tidak mengalami *overfitting* karena nilai *loss* untuk data latih dan untuk data validasi sama- sama menurun.



Gambar 4. Grafik hasil pada proses *training* model *Ensemble* (a) nilai akurasi (b) nilai *loss*

3.2. Tahap Pengujian

Pada tahap pengujian, bobot yang diperoleh dari proses pelatihan akan diuji menggunakan data yang tidak diuji sebelumnya dengan tujuan untuk melihat keberhasilan model *ensemble* dalam melakukan klasifikasi DR. untuk melihat nilai evaluasi dapat dilihat pada *confusion matrix* yang dapat dilihat pada Tabel 2.

Tabel 2. Hasil *confusion matrix* tahap pengujian *ensemble*

Kelas Aktual	Kelas Prediksi	
	Normal	DR
Normal	1275	530
DR	11	1846

Berdasarkan Tabel 2 diperoleh nilai $TP = 1275$ yang artinya jumlah prediksi normal diklasifikasikan sebagai normal sebanyak 1275. Nilai $FP = 11$ yang artinya jumlah prediksi DR tetapi diklasifikasikan sebagai normal sebanyak 11. Nilai $FN = 530$ yang artinya jumlah prediksi normal yang diklasifikasikan sebagai DR sebanyak 530. $TN = 1846$ jumlah prediksi DR yang diklasifikasikan sebagai DR sebanyak 1846.

Pada tahap ini akan menghitung nilai kinerja model *ensemble* berdasarkan akurasi, sensitivitas, spesifisitas, *f1-score* dan *cohens kappa*. Perhitungan yang dilakukan menggunakan Persamaan (12), (13), (14), (15) dan (16) berdasarkan *confusion matrix* pada Tabel 2. Sehingga diperoleh hasil nilai akurasi sebesar 85,22%, sensitivitas sebesar 70,63%, spesifisitas sebesar 99,40%, *f1-score* sebesar 87,21%, dan *cohens kappa* sebesar 70,32%.

3.3. Pembahasan

Pada penelitian ini telah dilakukan klasifikasi penyakit DR menggunakan metode *ensemble* pada arsitektur *Densenet*, *MobileNet* dan *Xception*. Untuk mengetahui keberhasilan model yang diusulkan untuk klasifikasi DR, maka hasil dari metode *ensemble* akan dibandingkan dengan klasifikasi tunggal. Perbandingan hasil klasifikasi antara klasifikasi tunggal dan multiclass dapat dilihat pada Tabel 3.

Tabel 3. Hasil perbandingan kinerja antara klasifikasi tunggal dan multiclass

Metode	Hasil Kinerja				
	Akurasi	Sensitivitas	Spesifisitas	F1-score	Cohens kappa
<i>DenseNet</i>	78,12%	79,11%	77,24%	77,3%	56,21%
<i>MobileNet</i>	84,98%	71,63%	97,95%	86,86%	69,84%
<i>Xception</i>	80,83%	61,55%	99,56%	84,04%	61,44%
<i>Ensemble</i>	85,22%	70,63%	99,40%	87,21%	70,32%

Pada Tabel 3. terlihat hasil akurasi menggunakan metode *ensemble* memiliki akurasi terbesar yaitu 85,22% dibandingkan ketiga arsitektur lainnya. Hal ini berarti metode *ensemble* kinerjanya baik dalam melakukan klasifikasi DR. Nilai sensitivitas pada metode *ensemble* memiliki nilai sebesar 70,63% namun hasil tertinggi terdapat pada arsitektur *DenseNet* dengan nilai sebesar 79,11%. Nilai spesifisitas pada metode *ensemble* memiliki nilai sebesar 99,40% namun hasil tertinggi terdapat pada arsitektur *Xception* dengan nilai sebesar 99,56%. *F1-score* pada metode *ensemble* memiliki nilai sebesar 87,21%, hasil ini merupakan hasil tertinggi dibandingkan arsitektur lainnya. Untuk nilai *cohens kappa* pada metode *ensemble* memiliki nilai sebesar 70,32%, hasil ini merupakan hasil tertinggi dibandingkan arsitektur lainnya. Adapun perbandingan hasil klasifikasi DR dengan penelitian lain dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan metode yang diusulkan dengan peneliti lain

Metode	Hasil Kinerja				
	Akurasi	Sensitivitas	Spesifisitas	F1-score	Cohens kappa
(<i>Inception V3</i> , <i>ResNet152</i> , <i>Inception-ResNet-V2</i>)	88,21%	84,35%	91,46%	-	-
Jiang <i>et al.</i> , [25]					
(<i>DenseNet121</i> , <i>Resnet50</i> , <i>Inception-V3</i>)	80,36%	47,70%	85,94%	-	-
Jinfeng <i>et al.</i> , [26]					
(VGG-16 dan VGG-19)	82%	80%	82%	-	-
Nguyen <i>et al.</i> [27]					
(<i>Inception V3</i> dan QIY model)	90%	-	-	-	-
Hagos and Kant [28]					
Metode yang diusulkan	85,22%	70,63%	99,40%	87,21%	70,32%

Berdasarkan Tabel 4 metode yang diusulkan memperoleh nilai spesifisitas, *f1-score*, dan *cohens kappa* yang tertinggi dari penelitian lainnya berturut-turut sebesar 99,40%, 87,21%, 0,7032. Untuk nilai akurasi tertinggi diperoleh oleh Hagos and Kant [25] sedangkan metode yang diusulkan memperoleh hasil akurasi sebesar 85,22%. Nilai sensitivitas tertinggi diperoleh oleh [25] sedangkan metode yang diusulkan memperoleh sensitivitas sebesar 70,63%. Penelitian yang dilakukan [25], [26], [27] hanya menghitung nilai kinerja akurasi, sensitivitas, dan spesifisitas tanpa menghitung nilai *f1-score* dan *cohens kappa*. Sedangkan penelitian [28] hanya menghitung nilai akurasi tanpa menghitung nilai sensitivitas, spesifisitas, *f1-score*, dan *cohens kappa*.

Berdasarkan Tabel 4, hasil evaluasi kinerja model *ensemble* yang diusulkan dapat disimpulkan bahwa keakuratan suatu model untuk melakukan klasifikasi penyakit DR sudah baik ditunjukkan dengan nilai akurasi sebesar 85,22%. Nilai sensitivitas yang dihasilkan sebesar 70,63% sudah cukup baik dalam dalam memprediksi label DR. nilai spesifisitas sebesar 99,40% sudah sangat baik dibandingkan peneliti lain yang artinya kemampuan model memprediksi label normal dengan benar sudah sangat baik. Untuk nilai *f1-score* yang diperoleh sebesar 87,21% artinya harmonisasi antara nilai sensitivitas dan spesifisitas sudah baik. Untuk nilai *cohens kappa* sudah baik yang artinya tingkat kesesuaian antara hasil prediksi yang dihasilkan dengan label sebenarnya sudah baik dengan nilai sebesar 70,32%.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan, penerapan metode *ensemble* dengan arsitektur *DenseNet*, *MobileNet*, dan *Xception* memperoleh hasil yang baik untuk melakukan klasifikasi DR. Nilai akurasi naik sampai 3,91% dibandingkan dengan rata-rata akurasi dari ketiga arsitektur yaitu 81,31%. Nilai sensitivitas pada metode *ensemble* sebesar 70,63% sedikit lebih kecil daripada rata-rata sensitivitas dari ketiga arsitektur yaitu 70,76%. Nilai spesifisitas naik sampai 7,82% dibandingkan dengan rata-rata spesifisitas dari ketiga arsitektur yaitu 91,58%. Nilai *f1-score* naik sampai 4,48% dibandingkan dengan rata-rata *f1-score* dari ketiga arsitektur yaitu 82,73%. Hasil kinerja klasifikasi tunggal *DenseNet*, *MobileNet*, dan *Xception* sudah cukup baik namun model pada klasifikasi tunggal pada saat pelatihan masih mengalami *overfitting*. Metode *ensemble* dapat mengatasi *overfitting* pada klasifikasi tunggal dengan hasil kinerja akurasi, spesifisitas, *f1-score* dan *cohens kappa* yang sudah baik meskipun nilai sensitivitas masih rendah. Nilai sensitivitas yang lebih rendah dibandingkan spesifisitas menunjukkan bahwa model cenderung lebih berhati-hati dalam mengklasifikasikan citra sebagai *Diabetic Retinopathy*. Hal ini kemungkinan disebabkan oleh karakteristik fitur yang dominan pada kelas normal serta strategi *ensemble* yang lebih mengutamakan konsistensi prediksi antar model. Dalam konteks klinis, spesifisitas yang sangat tinggi (99,40%) penting untuk meminimalkan kesalahan diagnosis pada pasien sehat. Penelitian ini masih memiliki beberapa keterbatasan, antara lain penggunaan dataset yang terbatas, skenario klasifikasi biner yang menyederhanakan kompleksitas klinis *Diabetic Retinopathy*, serta belum adanya validasi eksternal pada dataset dari institusi medis lain. Selain itu, metode *weighted voting* yang digunakan masih bersifat statis dan belum adaptif terhadap perubahan distribusi data. Metode yang diusulkan berpotensi dikembangkan lebih lanjut sebagai sistem pendukung keputusan untuk membantu dokter dalam proses *skrining* awal *Diabetic Retinopathy*.

DAFTAR PUSTAKA

- [1] S. Agarwal and A. Bhat, "A survey on recent developments in diabetic retinopathy detection through integration of deep learning," *Multimed. Tools Appl.*, vol. 82, no. 11, pp. 17321–17351, 2023.
- [2] I. R. I. Haque and J. Neubert, "Deep learning approaches to biomedical image segmentation," *Informatics Med. Unlocked*, vol. 18, p. 100297, 2020.
- [3] C. Liu, P. Gu, and Z. Xiao, "Multiscale U-Net with Spatial Positional Attention for Retinal Vessel Segmentation," *J. Healthc. Eng.*, vol. 2022, no. 1, p. 5188362, 2022.

- [4] S. Kaur and K. S. Mann, "Retinal vessel segmentation using an entropy-based optimization algorithm," *Int. J. Healthc. Inf. Syst. Informatics*, vol. 15, no. 2, pp. 61–79, 2020.
- [5] O. M. Al-hazimeh, A. A. Abu-Ein, N. M. Tahat, M. A. Al-Smadi, and M. M. Al-Nawashi, "Combining Artificial Intelligence and Image Processing for Diagnosing Diabetic Retinopathy in Retinal Fundus Images," *Int. J. Online Biomed. Eng.*, vol. 18, no. 13, 2022.
- [6] A. Samanta, A. Saha, S. C. Satapathy, S. L. Fernandes, and Y.-D. Zhang, "Automated detection of diabetic retinopathy using convolutional neural networks on a small dataset," *Pattern Recognit. Lett.*, vol. 135, pp. 293–298, 2020.
- [7] G. Ali, A. Dastgir, M. W. Iqbal, M. Anwar, and M. Faheem, "A hybrid convolutional neural network model for automatic diabetic retinopathy classification from fundus images," *IEEE J. Transl. Eng. Heal. Med.*, vol. 11, pp. 341–350, 2023.
- [8] Q. H. Nguyen *et al.*, "Diabetic retinopathy detection using deep learning," in *Proceedings of the 4th international conference on machine learning and soft computing*, 2020, pp. 103–107.
- [9] R. Biswas, A. Vasan, and S. S. Roy, "Dilated deep neural network for segmentation of retinal blood vessels in fundus images," *Iran. J. Sci. Technol. Trans. Electr. Eng.*, vol. 44, no. 1, pp. 505–518, 2020.
- [10] R. Sulthana, V. Chamola, Z. Hussain, F. Albalwy, and A. Hussain, "A novel end-to-end deep convolutional neural network based skin lesion classification framework," *Expert Syst. Appl.*, vol. 246, p. 123056, 2024.
- [11] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artif. Intell. Rev.*, vol. 57, no. 4, p. 99, 2024.
- [12] M. Rao, M. Zhu, and T. Wang, "Conversion and implementation of state-of-the-art deep learning algorithms for the classification of diabetic retinopathy," *arXiv Prepr. arXiv2010.11692*, 2020.
- [13] A. Souid, N. Sakli, and H. Sakli, "Classification and predictions of lung diseases from chest x-rays using mobilenet v2," *Appl. Sci.*, vol. 11, no. 6, p. 2751, 2021.
- [14] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, "Cancer image classification based on DenseNet model," in *Journal of physics: conference series*, 2020, p. 12143.
- [15] I. Kandel and M. Castelli, "Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review," *Appl. Sci.*, vol. 10, no. 6, p. 2021, 2020.
- [16] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.
- [17] A. Derry, M. Krzywinski, and N. Altman, "Convolutional neural networks," 2023, *Nature Publishing Group US New York*.
- [18] S. Albahli, T. Nazir, A. Irtaza, and A. Javed, "Recognition and Detection of Diabetic Retinopathy Using Densenet-65 Based Faster-RCNN," *Comput. Mater. Contin.*, vol. 67, no. 2, 2021.
- [19] N. Hasan, Y. Bao, A. Shawon, and Y. Huang, "DenseNet convolutional neural networks application for predicting COVID-19 using CT image," *SN Comput. Sci.*, vol. 2, no. 5, p. 389, 2021.
- [20] S. Mishra, S. Hanchate, and Z. Saquib, "Diabetic retinopathy detection using deep learning," in *2020 International conference on smart technologies in computing, electrical and electronics (ICSTCEE)*, 2020, pp. 515–520.
- [21] H. Liu, K. Yue, S. Cheng, C. Pan, J. Sun, and W. Li, "Hybrid model structure for diabetic retinopathy classification," *J. Healthc. Eng.*, vol. 2020, no. 1, p. 8840174, 2020.
- [22] S. Zhu, C. Xiong, Q. Zhong, and Y. Yao, "Diabetic retinopathy classification with deep learning via fundus images: A short survey," *IEEE Access*, vol. 12, pp. 20540–20558, 2024.
- [23] S. Bagui *et al.*, "Effect of hold-type on cyclic life and microstructural evolution of an austenitic stainless steel," *Materialia*, vol. 37, p. 102211, 2024.
- [24] T. Yumeng, C. Lina, and others, "Pneumonia Detection in chest X-rays: A deep learning approach based on ensemble RetinaNet and Mask R-CNN," in *2020 Eighth International Conference on Advanced Cloud and Big Data (CBD)*, 2020, pp. 213–218.
- [25] B. K. Chaurasia, H. Raj, S. S. Rathour, and P. B. Singh, "Transfer learning-driven ensemble model for detection of diabetic retinopathy disease," *Med. Biol. Eng. Comput.*, vol. 61, no. 8, pp. 2033–2049, 2023.
- [26] G. Jinfeng, S. Qummar, Z. Junming, Y. Ruxian, and F. G. Khan, "Ensemble framework of deep CNNs for diabetic retinopathy detection," *Comput. Intell. Neurosci.*, vol. 2020, no. 1, p. 8864698, 2020.
- [27] C. Zhang, T. Lei, and P. Chen, "Diabetic retinopathy grading by a source-free transfer learning approach," *Biomed. Signal Process. Control*, vol. 73, p. 103423, 2022.
- [28] R. Wardoyo, A. Musdholifah, G. A. Pradipta, and I. N. H. Sanjaya, "Weighted majority voting by statistical performance analysis on ensemble multiclassifier," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020, pp. 1–8.
- [29] W. Yu, S. Li, T. Ye, R. Xu, J. Song, and Y. Guo, "Deep ensemble machine learning framework for the estimation of PM2. 5 concentrations," *Environ. Health Perspect.*, vol. 130, no. 3, p. 37004, 2022.
- [30] D. Yang, G. Liu, M. Ren, B. Xu, and J. Wang, "A multi-scale feature fusion method based on u-net for retinal vessel segmentation," *Entropy*, vol. 22, no. 8, p. 811, 2020.
- [31] R. Shang, J. He, J. Wang, K. Xu, L. Jiao, and R. Stolkin, "Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification," *Knowledge-Based Syst.*, vol. 194, p. 105542, 2020.
- [32] W. K. Wong, F. H. Juwono, and B. T. T. Khoo, "Multi-features capacitive hand gesture recognition sensor: A machine learning approach," *IEEE Sens. J.*, vol. 21, no. 6, pp. 8441–8450, 2021.
- [33] K. Van Krieken and J. Sanders, "What is narrative journalism? A systematic review and an empirical agenda," *Journalism*, vol. 22, no. 6, pp. 1393–1412, 2021.
- [34] L. I. Kesuma and M. I. Anggara, "Implementation of Image Quality Improvement Methods and Lung Segmentation on Chest X-Ray Images Using U-Net Architectural Modifications," *Comput. Eng. Appl. J.*, vol. 12, no. 2, pp. 71–78, 2023.
- [35] G. C. Cardarilli *et al.*, "A pseudo-softmax function for hardware-based high speed image classification," *Sci. Rep.*, vol. 11, no. 1, p. 15307, 2021.

- [36] C. Chen, J. H. Chuah, R. Ali, and Y. Wang, "Retinal vessel segmentation using deep learning: a review," *IEEE Access*, vol. 9, pp. 111985–112004, 2021.
- [37] Y. Ye *et al.*, "Fully-automated segmentation of nasopharyngeal carcinoma on dual-sequence MRI using convolutional neural networks," *Front. Oncol.*, vol. 10, p. 166, 2020.
- [38] S. Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*, 2020, pp. 1–7.
- [39] K. Teeyapan, "Abnormality detection in musculoskeletal radiographs using EfficientNets," in *2020 24th international computer science and engineering conference (ICSEC)*, 2020, pp. 1–6.
- [40] F. Stahlberg, "Neural machine translation: A review," *J. Artif. Intell. Res.*, vol. 69, pp. 343–418, 2020.
- [41] Y. Zuo, Q. Zou, J. Lin, M. Jiang, and X. Liu, "2lpiRNAPred: a two-layered integrated algorithm for identifying piRNAs and their functions based on LFE-GM feature selection," *RNA Biol.*, vol. 17, no. 6, pp. 892–902, 2020.
- [42] T. Schopf, D. Braun, and F. Matthes, "Evaluating unsupervised text classification: zero-shot and similarity-based approaches," in *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, 2022, pp. 6–15.
- [43] Y. Ma, S. Liu, Z. Li, and J. Sun, "Iqdet: Instance-wise quality distribution sampling for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1717–1725.
- [44] M. M. Musthafa, M. TR, V. K. V, and S. Guluwadi, "Enhanced skin cancer diagnosis using optimized CNN architecture and checkpoints for automated dermatological lesion classification," *BMC Med. Imaging*, vol. 24, no. 1, p. 201, 2024.
- [45] D. J. Hand, P. Christen, and S. Ziyad, "Selecting a classification performance measure: matching the measure to the problem," *arXiv Prepr. arXiv2409.12391*, 2024.
- [46] D. Chicco, M. J. Warrens, and G. Jurman, "The Matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and Brier score in binary classification assessment," *Ieee Access*, vol. 9, pp. 78368–78381, 2021.
- [47] S. S. Mullick, S. Datta, S. G. Dhekane, and S. Das, "Appropriateness of performance indices for imbalanced data classification: An analysis," *Pattern Recognit.*, vol. 102, p. 107197, 2020.