

CATARACT CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK (CNN) INCEPTION RESNETV2

M. Mauludin Zulfa^{*1}, Christian Sri Kusuma Aditya²

^{1,2}Informatics, Engineering Faculty, University of Muhammadiyah Malang, Indonesia
Email: mauludinzulfa@webmail.umm.ac.id, christianskaditya@umm.ac.id

(Article received: June 7, 2024; Revision: July 29, 2024; published: October 20, 2024)

Abstract

The eye is a human sensory device that functions as an organ of vision. Referring to data from the World Health Organization (WHO) in 2018, cataracts are responsible for 48% of blindness cases in the world and are the main cause in Indonesia. People still find it difficult to distinguish cataract eyes from normal eyes, so they often do not realize the indications of cataract disease. It is important to conduct early detection of cataract disease before blindness occurs. As technology develops, cataract identification becomes easier and simpler with digital image processing classification. This research develops a cataract image classification model using Convolutional Neural Network (CNN) with Inception-ResnetV2 architecture to identify cataract eyes with normal eyes. The proposed model consists of two parts of Inception-ResnetV2 architecture as the base model, and the head model in the form of Fully Connected Layers consisting of global average pooling, 2 dense relu layers of 128 and 256 neurons, 2 batch normalization layers, 2 layers of dropout parameter 0.5, and softmax activation function for the output layer. To improve model training, the Stochastic Gradient Descent (SGD) optimization function is used. The dataset consists of 2,192 eye fundus images with 2 main classes of cataract and normal taken from the public data provider site Kaggle. Learning rate tests on the optimization function were carried out with parameters 0.1, 0.01, and 0.001. The results of the proposed model compiled with Stochastic Gradient Descent (SGD) learning rate 0.01 gave a final accuracy of 96%.

Keywords: Cataract Classification, CNN, Inception-ResNetV2, Stochastic Gradient Descent (SGD)

KLASIFIKASI KATARAK MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) ARSITEKTUR INCEPTION RESNETV2

Abstrak

Mata merupakan alat indera manusia yang berfungsi sebagai organ pengelihatan. Mengacu pada data *World Health Organization* (WHO) tahun 2018, katarak bertanggung jawab atas 48% kasus kebutaan di dunia dan menjadi penyebab utama di Indonesia. Masyarakat masih sulit membedakan mata katarak dengan mata normal, sehingga sering tidak menyadari terjadinya indikasi penyakit katarak. Penting untuk melakukan pendeteksian dini penyakit katarak sebelum mengalami kebutaan. Seiring berkembangnya teknologi, identifikasi katarak menjadi lebih mudah dan sederhana dengan adanya klasifikasi pengolahan citra digital. Penelitian ini mengembangkan model klasifikasi citra katarak menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *Inception-ResnetV2* untuk mengidentifikasi mata katarak dengan mata normal. Model yang diusulkan terdiri dari dua bagian arsitektur *Inception-ResnetV2* sebagai *base model*, dan *head model* berupa lapisan *Fully Connected* terdiri dari *global average pooling*, 2 lapisan *dense ReLU* 128 dan 256 neuron, 2 lapisan *normalisasi batch*, 2 lapisan *dropout* parameter 0,5, dan fungsi aktivasi *softmax* untuk lapisan *output*. Untuk meningkatkan pelatihan model, fungsi optimasi *Stochastic Gradient Descent* (SGD) digunakan. Dataset terdiri dari 2.192 gambar fundus mata dengan 2 kelas utama katarak dan normal diambil dari situs penyedia data publik *kaggle*. Pengujian *learning rate* pada fungsi optimasi dilakukan dengan parameter 0,1, 0,01, dan 0.001, hasilnya model usulan dikompilasi dengan *Stochastic Gradient Descent* (SGD) *learning rate* 0,01 memberikan akurasi akhir sebesar 96%.

Kata kunci: Klasifikasi Katarak, CNN, Inception-ResNetV2, Stochastic Gradient Descent (SGD)

1. PENDAHULUAN

Mata merupakan organ indera yang berperan dalam penglihatan dan merupakan bagian penting dari tubuh manusia. Secara teratur, mata mengubah

cahaya pada objek untuk menghasilkan gambar dan dikirim ke otak [1]. Lebih dari 50 jenis penyakit dapat mempengaruhi fungsi mata, kebutaan menjadi salah satu gangguan penglihatan yang paling serius. Menurut data *World Health Organization* (WHO) tahun 2018, katarak menjadi penyebab sekitar 48% kasus kebutaan global dan merupakan penyebab utama kebutaan di Indonesia. [2]. Menurut Riset Kesehatan Dasar (Riskesdas 2018), 2,5% penduduk Indonesia mengalami kebutaan akibat katarak. Angka ini menjadikan Indonesia sebagai negara dengan kasus katarak tertinggi di Asia, menyumbang 78% dari semua kasus di Asia Tenggara [3]. Katarak didefinisikan sebagai kondisi kerusakan pada lensa mata akibat penambahan cairan (hidrasi lensa), kondisi ini menyebabkan penurunan penglihatan yang signifikan bahkan kehilangan penglihatan secara keseluruhan [4]. Penyebab katarak meliputi faktor idiopatik, genetik, kelainan mata, sindrom multisistem, kelainan metabolisme, infeksi selama kehamilan, paparan radiasi, dan cedera pada mata. Faktor risiko terjadinya katarak antara lain yaitu diabetes, merokok, indeks massa tubuh, dan refraksi [5].

Strategi pengobatan paling efektif pada katarak saat ini yaitu operasi yang dikombinasikan dengan implantasi lensa [6]. Operasi mata dapat menyebabkan berbagai komplikasi yang berdampak pada kualitas penglihatan. Beberapa komplikasi umum yang terjadi seperti edema makula dan peningkatan tekanan intra okular (*glaucoma*) [7]. Untuk meminimalkan resiko, dokter spesialis mata merupakan pakar yang tepat untuk menghindari komplikasi serius, Identifikasi yang cepat dan akurat sangat penting dalam menangani masalah pascaoperasi [8]. Saat ini, pengolahan citra digital telah mempermudah dan menyederhanakan proses deteksi dan klasifikasi penyakit katarak [2]. Teknologi ini digunakan sebagai parameter untuk menghasilkan pola yang kemudian dibagi menjadi beberapa kategori yang ditentukan. Proses klasifikasi atau identifikasi citra banyak diaplikasikan menggunakan teknik *deep learning* (DL) seperti klasifikasi *neural network*.

Deep learning (DL) merupakan salah satu bidang dari *machine learning* yang algoritmanya terinspirasi dari struktur otak manusia [9]. *Convolutional Neural Network* (CNN) merupakan jenis metode *deep learning* yang paling populer untuk klasifikasi citra. CNN menggunakan data 2D yang cocok untuk pemrosesan gambar, serta perhitungan konvolusi untuk matriks dan bobot 4 dimensi [10]. Pada dasarnya, CNN merupakan evolusi dari *Artificial Neural Network* (ANN) [11]. Karena kinerjanya yang sangat baik, CNN sering digunakan dalam pendekatan kecerdasan buatan untuk memecahkan masalah klasifikasi citra [12]. Struktur CNN terdiri dari *input*, ekstraksi fitur, proses klasifikasi, dan *output*. Proses ekstraksi CNN terdiri dari beberapa lapisan tersembunyi atau *hidden*

layer, termasuk lapisan konvolusi, *pooling*, dan fungsi aktivasi *ReLU* (*Rectification Linear Unit*) [13]. CNN bekerja secara hierarki dalam arti, *output* dari lapisan konvolusi pertama digunakan sebagai input untuk lapisan konvolusi berikutnya. Dalam proses klasifikasi, terdapat *fully-connected* dan fungsi aktivasi (*softmax*) dengan output berupa hasil klasifikasi [14].

Inception-ResNetV2 merupakan arsitektur CNN yang dikembangkan dari keluarga arsitektur Inception dengan memasukkan elemen koneksi residual, menggantikan serangkaian filter yang ada pada arsitektur Inception [13]. *Residual Network* (*ResNet*) merupakan *pre trained model* atau model yang telah dilatih sebelumnya sehingga konfigurasi khusus untuk mengatur layer di dalamnya tidak diperlukan [15]. Arsitektur *inception-ResNetV2* telah menarik perhatian karena kinerjanya untuk mengekstrak fitur dari citra dengan kedalaman yang luar biasa [13]. *Inception* membuat arsitektur yang cenderung melebar dari pada jaringan yang mendalam untuk menghindari layer yang semakin dalam. Penumpukan layer yang dalam merupakan hasil dari banyak pengembangan arsitektur CNN. *Overfitting* dan *vanishing gradien* dengan lapisan yang semakin dalam tidak menyebabkan tingkat kesalahan berkurang tetapi meningkat, sehingga menyebabkan waktu komputasi yang lama [16].

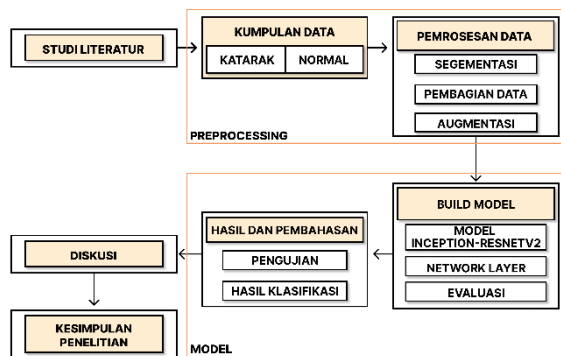
Penelitian terdahulu telah membuktikan keberhasilan terkait penggunaan teknik *deep learning* (DL) dengan metode berbasis jaringan saraf konvolusional (CNN) untuk klasifikasi citra. Penelitian yang dilakukan oleh E. Andreas dan W. Widhiarso (2020) menggunakan CNN dengan model *inceptionV3* optimasi *adam* dan augmentasi data menghasilkan nilai akurasi 100%, namun memiliki risiko *overfitting* yang tinggi terutama jika dataset tidak cukup beragam [17]. Studi oleh E. G. Winarto dan A. Lawi (2021), menggunakan arsitektur CNN dengan *ResNetV2* untuk klasifikasi citra biji kakao dan mencapai akurasi sebesar 89% dan ROC 97% [13]. Faktor seperti jumlah layer, ukuran kernel layer input, ketepatan parameter augmentasi data, serta keseimbangan data mempengaruhi hasil klasifikasi. Perbandingan kinerja berbagai arsitektur *deep learning* juga telah dilakukan oleh F. Masrurroh, B. Surarso, dan B. Warsito [18]. Penelitian ini membandingkan empat arsitektur pada gambar wajah yaitu *Xception*, *ResNet50*, *InceptionResNet-v2*, dan *Inception-v3*. Hasil penelitian menunjukkan bahwa arsitektur *Inception-ResNet-V2* dengan fungsi aktivasi *Mish* dan *optimizer Nadam* mencapai nilai akhir *accuracy* dan *f1-score* sebesar 92%. Meskipun konfigurasi ini lebih kompleks dalam hal tuning parameter.

Berdasarkan studi-studi terdahulu, penelitian ini menggunakan arsitektur *Inception-ResNet-V2* dengan optimasi *Stochastic Gradient Descent* (SGD) dan aktivasi *Softmax* dengan penyesuaian *hyperparameter* yang bertujuan untuk meningkatkan

keseimbangan antara kompleksitas dan kinerja untuk membangun sistem pakar yang dapat mengklasifikasikan citra katarak. Penelitian ini diharapkan dapat mengatasi kelemahan dalam penelitian sebelumnya dan memberikan hasil yang lebih andal dalam deteksi katarak.

2. METODE PENELITIAN

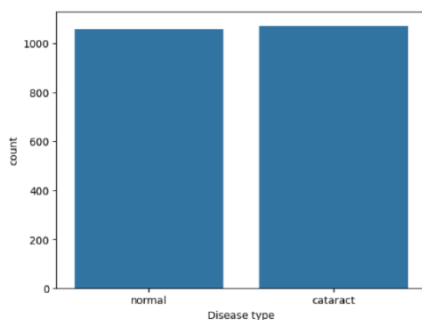
Dalam penelitian ini, digunakan arsitektur *Convolutional Neural Network (CNN) Inception-ResNet-V2* dengan bahasa pemrograman Python 3 serta library Keras dan TensorFlow untuk mengembangkan dan melatih model. Pengembangan dilakukan pada *Integrated Development Environment (IDE) Visual Studio Code (VSCode)*. Alur metodologi penelitian ini diilustrasikan pada Gambar 1.



Gambar 1. Alur Metodologi Penelitian

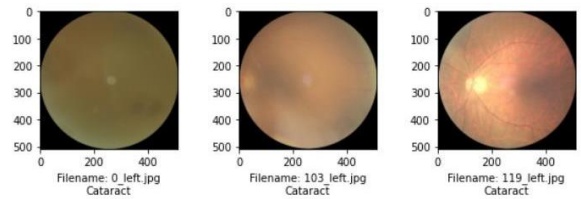
2.1. Kumpulan Data

Kumpulan dataset penelitian ini merupakan gambar fundus mata dari *kaggle*. *Kaggle* merupakan sebuah *platform* yang dikenal sebagai bank data publik dan menjadi wadah bagi kompetisi ilmu data. Selain itu, *kaggle* juga menyediakan beragam dataset serta lingkungan *cloud* untuk keperluan ilmu data [19]. Dataset penelitian ini berjumlah 2.192 gambar fundus mata dengan pembagian 1.119 gambar fundus kelas katarak dan 1.073 gambar fundus kelas normal. Dataset yang baik merupakan fondasi yang kuat dalam membangun model, serta menguji keandalan algoritma yang dikembangkan untuk menjamin kepastian hasil penelitian [20].



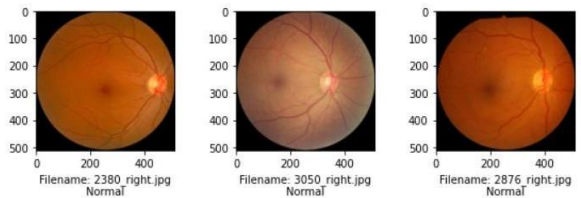
Gambar 2. Distribusi Dataset

Gambar 2 merupakan tampilan distribusi jumlah data pada kedua kelas, diagram ditampilkan menggunakan *countplot*.



Gambar 3. Citra Fundus Kelas Katarak

Gambar 3 menunjukkan katarak pada citra ditandai oleh kehilangan transparansi dan kekeruhan pada lensa mata. Kondisi ini menyebabkan lensa mata menjadi buram dan menyebarkan cahaya saat melewatinya, yang mengurangi ketajaman penglihatan di retina dan menyebabkan penglihatan menjadi kabur [21].

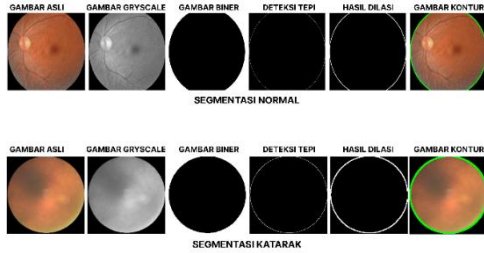


Gambar 4. Citra Fundus Kelas Normal

Pada gambar 4 mata normal didefinisikan sebagai keadaan dimana mata dalam kondisi baik, lensa mata yang jernih dan dilalui cahaya menuju retina [22].

2.2. Pemrosesan Data

Tahap persiapan atau *preprocessing*, yaitu proses yang dilakukan sebelum fase pelatihan untuk mengubah data mentah menjadi format yang siap, terstruktur, dan dapat digunakan. Langkah-langkah dalam tahap ini meliputi segmentasi gambar yang dimulai dengan membaca gambar menggunakan *cv2.imread*, lalu dikonversi ke *grayscale* dengan *cv2.cvtColor*. Metode *Otsu* diterapkan menggunakan *cv2.threshold* untuk menghasilkan gambar biner, diikuti dengan deteksi tepi menggunakan *cv2.Canny*. Hasil deteksi tepi dilasi menggunakan kernel 3x3 dengan *cv2.dilate*. Kontur ditemukan dengan *cv2.findContours* dan diurutkan menggunakan *imutils.sort_contours*. Kontur yang ditemukan digambar pada gambar asli dengan *cv2.drawContours* dan ditampilkan menggunakan *Matplotlib* dengan *plt.imshow*. Kontur yang dihasilkan siap untuk pemrosesan lebih lanjut. Untuk detail gambar sampel citra hasil segmentasi dapat dilihat pada Gambar 5.



Gambar 5. Sampel Citra Hasil Segmentasi

Selanjutnya dilakukan *resize* gambar dengan target 100 x 100 *pixel* dan membagi keseluruhan dataset ke dalam 3 folder, pemisahan terdiri dari data latih (72%), data uji (4%), dan data validasi (24%). Setiap map terdiri dari dua kelas yaitu kelas normal dan kelas katarak. Untuk detail distribusi pembagian dataset dapat dilihat pada Tabel 1.

Tabel 1. Distribusi Pembagian Dataset

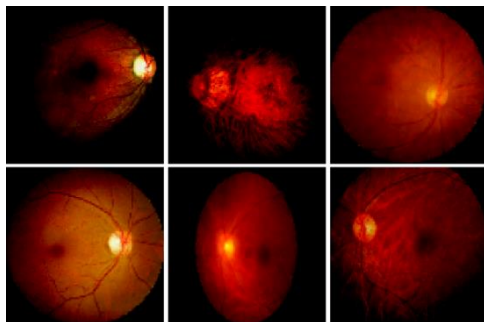
| FOLDER | JUMLAH |
|---------------|--------|
| DATA LATIH | 1584 |
| DATA UJI | 79 |
| DATA VALIDASI | 528 |

Dilakukan augmentasi data menggunakan *Image DataGenerator* dari *library Keras*. Dengan tujuan memberikan variasi tambahan kedalam data pelatihan [23]. Meningkatkan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya [24]. Parameter augmentasi yang digunakan pada penelitian ini dapat dilihat pada Tabel 2.

Tabel 2. Parameter Augmentasi

| Augmentasi | Parameter |
|-----------------|-----------|
| Rescale | 1.0/255 |
| Rotation_Range | 40 |
| Horizontal_Flip | True |
| Validation_Flip | 0,25 |

Geser, putar, atau membalik gambar dapat membantu model untuk mempelajari pola-pola dalam data yang lebih umum dan fleksibel. Melalui penerapan serangkaian teknik augmentasi dengan parameter-parameter yang telah ditentukan, model dapat dilatih dengan beragam variasi dari data asli [25].



Gambar 6. Citra Hasil Augmentasi

Gambar 6 merupakan sampel data latih yang telah ter-augmentasi. Hasilnya didapatkan variasi

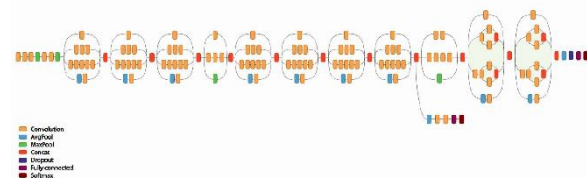
gambar yang beragam, dengan tujuan meningkatkan pelatihan model.

2.3. Build Model

Dalam pembuatan model, peneliti menggunakan metode *transfer learning*. Model telah dilatih sebelumnya (*pre-trained model*) yang digunakan adalah *Inception-ResNetV2* sebagai *base model*, ditambahkan arsitektur layer baru pada bagian *Fully Connected Layer* sebagai *head model*. Pada model yang dikompilasi, fungsi optimasi *Stochastic Gradient Descent* (SGD) diletakkan dengan fungsi *loss* menggunakan *binary_crossentropy*. Presisi, *recall*, *F1-Score*, dan akurasi dari *confusion matrix* digunakan untuk evaluasi hasil klasifikasi model yang dibangun.

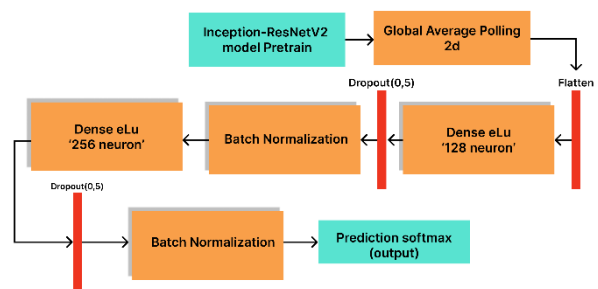
2.3.1. Model Inception-ResNetV2

Arsitektur *Inception-ResNet-V2* merupakan evolusi dari *Inception ResNet-V1* yang sukses pada lomba *ImageNetLarge Scale Visual Recognition Challenge* (ILSVRC) 2015, dan pendahulunya, *GoogLeNet*, yang memenangkan ILSVRC 2014 [13]. Pada tahap membangun model, penggunaan layer-layer khusus memiliki peran penting dalam memastikan model dapat belajar dan mengekstrak pola-pola yang kompleks dari data [26]. Terdapat serangkaian layer yang bertujuan untuk mengatur dan mengoptimalkan informasi yang mengalir di dalam model. Gambar 7 merupakan model arsitektur *Inception-ResNetV2*.



Gambar 7. Arsitektur Model Inception-ResNetV2

2.3.2. Network Layer



Gambar 8. Skema Network Layer Usulan

Pada Gambar 8 menunjukkan skema network layer usulan. Dimana pada tahapan selanjutnya, terdapat layer yang terkoneksi penuh (*Fully-connected layers*) digunakan untuk menyatukan informasi dari layer dalam pembentukan klasifikasi dimana neuron yang aktif pada satu lapisan terhubung dengan neuron pada lapisan selanjutnya [27]. Layer-layer ini memungkinkan model mendapatkan prediksi yang tepat. Pada penelitian ini layer yang digunakan terdiri dari 2 *layer dense ReLu* dengan parameter 128 dan 256 *neuron*. 2 *layer dropout* dengan parameter 0,5 dalam layer model berperan penting dalam mencegah *overfitting* dengan secara acak menonaktifkan sebagian unit (*neuron*) pada setiap *iterasi*, membantu model menjadi lebih tahan terhadap kecenderungan mengingat data latih secara berlebihan [28]. 2 *layer batch normalization* untuk mengontrol distribusi nilai-nilai di setiap layer, mengurangi masalah *vanishing* atau *exploding gradients* selama pelatihan dan membantu konvergensi model [29]. *Layer output* menggunakan fungsi aktivasi *softmax* untuk klasifikasi *multi-kelas*, mengonversi *output* model menjadi distribusi probabilitas. Penggunaan layer-layer ini secara bersama-sama membentuk sebuah klasifikasi yang stabil dan dapat diandalkan [30].

2.3.3. Evaluasi

Hasil klasifikasi model akan dilakukan evaluasi untuk menganalisa kinerja. Evaluasi dilakukan menggunakan rumus presisi, *recall*, *F1 score*, dan akurasi dari *confusion matrix*. Rumus untuk mendapatkan hasil evaluasi model dapat dilihat pada persamaan (2), (3), (4), dan (5).

$$Presisi = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1Score = \frac{2x(RecallxPresisi)}{(Recall + Presisi)} \tag{3}$$

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

| | | Actual Values | |
|------------------|--------------|--|---|
| | | 1 (Positive) | 0 (Negative) |
| Predicted Values | 1 (Positive) | <p>TP (True Positive)</p> | <p>FP (False Positive) <i>Type I Error</i></p> |
| | 0 (Negative) | <p>FN (False Negative) <i>Type II Error</i></p> | <p>TN (True Negative)</p> |

Gambar 9. Tabel *Confusion matrix*

Pada Gambar 9 menunjukkan Tabel *Confusion Matrix*, dimana tabel tersebut digunakan untuk menyatakan hasil klasifikasi dari prediksi data [31]. (TP) dan (TN) menggambarkan seberapa akurat model dalam mengklasifikasikan data, sedangkan (FP) dan (FN) memberikan gambaran tentang kesalahan yang dibuat model.

3. HASIL DAN PEMBAHASAN

Bahasa pemrograman *Python* versi 3.10.7 digunakan dalam penelitian ini dan dijalankan menggunakan *Visual Studio Code*. Pilihan alat dan bahasa ini didasarkan pada kemudahan penggunaan, fleksibilitas, dan dukungan komunitas yang luas. *Python* memiliki berbagai *library* untuk proses pembelajaran mesin dan analisis data, sementara *Visual Studio Code* menawarkan lingkungan pengembangan yang mudah digunakan dan efektif.

3.1. Pengujian

Stochastic Gradient Descent (SGD) adalah fungsi optimasi yang digunakan untuk meningkatkan proses pembelajaran model. Peneliti menguji kinerja optimasi dengan *learning rate* 0.1, 0.01, dan 0.001, selama 15 *epoch*, bertujuan menentukan *learning rate* yang memberikan performa terbaik bagi model pada jumlah iterasi yang sama.

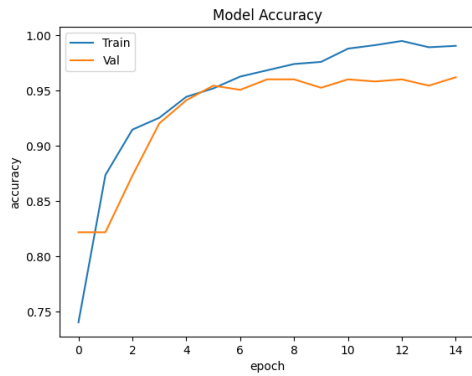
Tabel 3. Hasil Pengujian Learning Rate

| Fungsi Optimasi | Learning Rate | Epoch | Akurasi |
|-----------------|---------------|-------|---------|
| SGD | 0.1 | 15 | 0.95% |
| | 0.01 | | 0.96% |
| | 0.001 | | 0.94% |

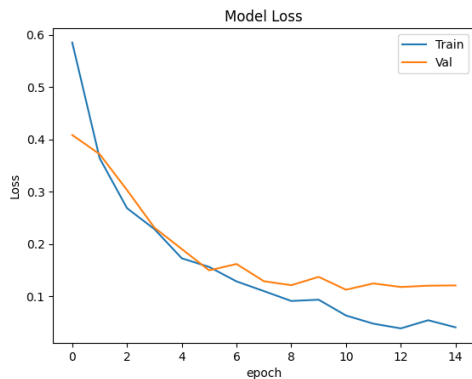
Hasil pengujian disajikan pada tabel 3, menunjukkan bahwa *Stochastic Gradient Descent* (SGD) memiliki nilai akurasi dan performa stabil selama pengujian. Tetapi, dengan *learning rate* 0.001, SGD menunjukkan akurasi yang lebih rendah banding *learning rate* lainnya, juga membutuhkan waktu komputasi yang lebih lama karena nilai *rate* terlalu kecil. Hasil tertinggi didapatkan pada *learning rate* 0.01. Membuktikan bahwa *learning rate* standar SGD adalah 0.01 [32].

3.2. Evaluasi Hasil Klasifikasi

Hasil ini didapatkan pada pengujian sebelumnya, model yang telah dibangun dikompilasi dengan fungsi optimasi *Stochastic Gradient Descent* (SGD) *learning rate* 0.01.



Gambar 10. Grafik Akurasi Model



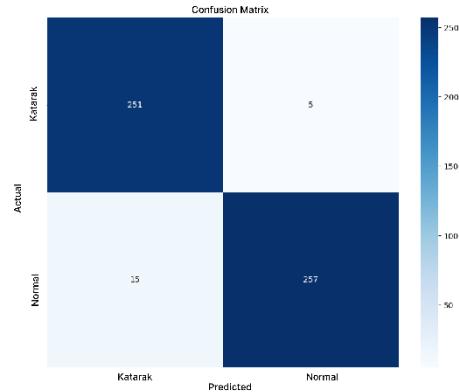
Gambar 11. Grafik Loss Model

Grafik pada Gambar 10 dan 11 menunjukkan model belajar dengan baik dari data pelatihan dan secara bertahap mengurangi kesalahan prediksi, menghasilkan akurasi pelatihan meningkat dan *loss* pelatihan memiliki tren penurunan yang signifikan hingga periode terakhir. Akurasi validasi mencapai nilai konsisten sebesar 0.96%, artinya model mampu menangkap pola yang relegated. Sebagai hasil akhir, didapatkan akurasi sebesar 96%.

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| cataract | 0.94 | 0.98 | 0.96 | 256 |
| normal | 0.98 | 0.94 | 0.96 | 272 |
| accuracy | | | 0.96 | 528 |
| macro avg | 0.96 | 0.96 | 0.96 | 528 |
| weighted avg | 0.96 | 0.96 | 0.96 | 528 |

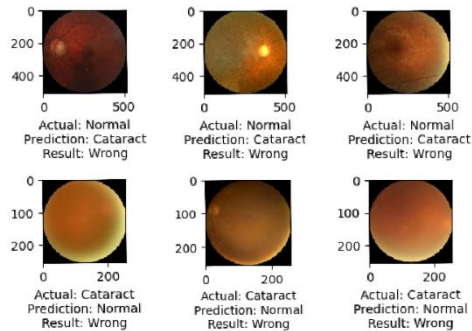
Gambar 12. Report Klasifikasi

Analisis *classification report* pada Gambar 12. Nilai akurasi mengukur seberapa banyak prediksi benar dari total prediksi. Jika akurasi tinggi tapi presisi dan recall rendah, model baik dalam memprediksi kelas mayoritas tetapi buruk dalam mengidentifikasi kelas minoritas pada data validasi. Presisi mengukur prediksi positif benar dari total prediksi positif, presisi rendah artinya banyak prediksi positif salah. Model bisa memiliki akurasi tinggi tetapi presisi dan *recall* rendah, penting mempertimbangkan metrik seperti *F1-score*, presisi, atau *recall*. Untuk gambaran kinerja yang kompleks.



Gambar 13. Confusion Matrix

Pada gambar 13 *confusion matrix* hasil klasifikasi dari total 528 data validasi, model menunjukkan adanya kesalahan prediksi menyebabkan 15 gambar pada kelas normal diprediksi sebagai katarak, dan 5 sampel gambar kelas katarak salah diklasifikasikan oleh model sebagai normal.



Gambar 14. Kesalahan Prediksi Model

Sampel dari total 20 gambar untuk kedua kelas yang salah diprediksi oleh model ditunjukkan pada Gambar 14. 3 gambar untuk setiap kelas digunakan untuk tujuan dokumentasi.

4. DISKUSI

Berdasarkan hasil yang telah didapatkan oleh peneliti, penggunaan model *Convolutional Neural Network* (CNN) arsitektur *Inception-ResNetV2* sebagai *base model* dengan penambahan layer pada *head model*. Kompilasi fungsi optimasi *Stochastic Gradient Descent* (SGD) *learning rate* 0.01, memberikan dampak positif pada hasil. Pengujian *learning rate* dengan parameter 0.1, 0.01, 0.001 pada model yang diusulkan memiliki tingkatan hasil akurasi dan *loss* yang berbeda, sehingga perlu diulas lebih lanjut untuk penggunaan layer tambahan, fungsi optimasi dan *learning rate* yang tepat untuk meningkatkan akurasi model yang dibangun.

Tabel 4. Penelitian Terdahulu

| Peneliti | Model | Optimasi | Dataset | Akurasi | Catatan |
|--|---|----------|--------------------|---------|--|
| E. Andreas dan W. Widhiarso (2020) | Inception V3 | Adam | Katarak | 100% | risiko <i>overfitting</i> yang tinggi terutama jika dataset tidak cukup beragam |
| E. G. Winarto dan A. Lawi (2021) | ResNetV2 | - | Biji Kakao | 89% | Faktor seperti jumlah layer, ukuran kernel, parameter augmentasi, dan keseimbangan data mempengaruhi hasil |
| F. Masruroh, B. Surarsolo, dan B. Warsito (2021) | <i>Xception</i> , <i>ResNet50</i> , <i>InceptionR</i> , <i>esNet-v2</i> , dan <i>Inception-v3</i> | Nadam | Gambar Bentu Wajah | 92% | Fungsi aktivasi Mish, tuning parameter kompleks |

Berdasarkan Tabel 4 yang menjelaskan penelitian – penelitian sebelumnya. Pada penelitian ini membandingkan beberapa arsitektur deep learning untuk klasifikasi citra dengan fokus pada akurasi dan risiko *overfitting*, serta menyoroiti pentingnya konfigurasi model dan parameter dalam mencapai hasil yang optimal.

5. KESIMPULAN

Penelitian ini, berhasil membuktikan metode berbasis *Convolutional Neural Network* (CNN) arsitektur *Inception-ResNetV2* dapat digunakan untuk mengklasifikasikan katarak pada gambar fundus. Dengan mengusulkan model *transfer learning*, model ini terdiri dari dua bagian yaitu arsitektur dasar *Inception-ResNetV2* dan arsitektur *head model* yang tersusun dari lapisan *global average pooling* diikuti lapisan *fully-connected layers*. Berdasarkan hasil eksperimen, model usulan dapat memberikan akurasi yang tinggi setelah dikompilasi dengan fungsi optimasi *Stochastic Gradient Descent* (SGD) *learning rate* 0.01. Meningkatkan kinerja model memberikan dampak positif dalam melakukan klasifikasi citra, memberikan prediksi akurat dalam mengidentifikasi katarak. Untuk penelitian lebih lanjut, dapat dikembangkan model untuk meningkatkan kinerja klasifikasi dan prediksi penyakit mata atau model yang dapat menemukan jenis penyakit mata tertentu.

DAFTAR PUSTAKA

- [1] M. Muafi, A. Wijaya, and V. A. Aziz, "Sistem Pakar Mendiagnosa Penyakit Mata Pada Manusia Menggunakan Metode Forward Chaining," *COREAI J. Kecerdasan Buatan, Komputasi dan Teknol. Inf.*, vol. 1, no. 1, pp. 43–49, 2020, doi: 10.33650/coreai.v1i1.1669.
- [2] L. D. B. Diki Hananta Firdaus, Bahtiar Imran and E. Suryadi, "Klasifikasi Penyakit Katarak Pada Mata Menggunakan Metode Convolutional Neural Network (Cnn) Berbasis Web Web-Based Classification of Cataract in the Eyes Using Convolutional Neural Network (Cnn) Method," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 1, no. 3, pp. 18–26, 2022.
- [3] S. Manovita Pateda and P. Ayuningtias Mahdang, "Pendampingan Kader Kesehatan dalam Melakukan Deteksi Dini Penyakit Katarak dengan Metode 'LIHAT,'" *J. Kolaboratif Sains*, vol. 6, no. 9, pp. 1133–1137, 2023, doi: 10.56338/jks.v6i9.4126.
- [4] P. Sisy Rizkia, "Jurnal Penelitian Perawat Profesional Pencegahan Tetanus," *Br. Med. J.*, vol. 2, no. 5474, pp. 1333–1336, 2020.
- [5] R. Dwi Hasriani, D. Pencegahan dan Pengendalian Penyakit Tidak Menular, and K. R. Kesehatan, "HIGEIA JOURNAL OF PUBLIC HEALTH RESEARCH AND DEVELOPMENT Hipertensi dengan Katarak pada Peserta Skrining Gangguan Penglihatan," vol. 4, no. 4, pp. 645–655, 2020, [Online]. Available: <http://journal.unnes.ac.id/sju/index.php/higeiahttps://doi.org/10.15294/higeia/v4i4/38745>
- [6] Sisy Rizkia Putri, "Jurnal Penelitian Perawat Profesional Pencegahan Tetanus," *Br. Med. J.*, vol. 2, no. 5474, pp. 1333–1336, 2020.
- [7] R. A. Nurjanah, S. N. Indawaty, and M. Purwoko, "Faktor Risiko Timbulnya Low Vision Pasca Operasi Ekstrakapsular Avoidable tajam penglihatan hingga mencapai untuk Hasil," *Syifa`MEDIKA*, vol. 10, no. 1, 2019.
- [8] Z. Farahdiva and A. Maisarah, "Screening Awal Penyakit Katarak dengan Image Processing menggunakan Metode Convolutional Neural Network pada Penyakit Akibat Kerja Pengelasan," no. 2581, 2023.
- [9] Rendi Nurcahyo and Mohammad Iqbal, "Pengenalan Emosi Pembicara Menggunakan Convolutional Neural Networks," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 1, pp. 115–122, 2022, doi: 10.29207/resti.v6i1.3726.

- [10] V. B. Gianzurriell, F. Husnal, F. A. Wijaya, F. Fauzi, I. Paryudi, and I. Veritawati, "Analisis Gambar mri otak untuk mendeteksi tumor otak menggunakan algoritma cmn," vol. 4, no. 2, pp. 14–18, 2023.
- [11] Bambang Pilu Hartato, "Penerapan Convolutional Neural Network pada Citra Rontgen Paru-Paru untuk Deteksi SARS-CoV-2," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 747–759, 2021, doi: 10.29207/resti.v5i4.3153.
- [12] D. Marcella, Y. Yohannes, and S. Devella, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur VGG-19," *J. Algoritm.*, vol. 3, no. 1, pp. 60–70, 2022, doi: 10.35957/algoritm.v3i1.3331.
- [13] E. G. Winarto and A. Lawi, "Implementasi Arsitektur Inception Resnet-V2 untuk Klasifikasi Kualitas Biji Kakao," *Konf. Nas. Ilmu Komput.*, pp. 132–137, 2021.
- [14] H. F. Jessar, A. T. Wibowo, and E. Rachmawati, "Klasifikasi Genus Tanaman Sukulen Menggunakan Convolutional Neural Network," vol. 8, no. 2, pp. 3180–3196, 2021.
- [15] A. Ridhovan and A. Suharso, "Penerapan Metode Residual Network (Resnet) Dalam Klasifikasi Penyakit Pada Daun Gandum," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 1, pp. 58–65, 2022, doi: 10.29100/jupi.v7i1.2410.
- [16] A. Satyo, B. Karno, D. Arif, I. Sari, and K. Wardhana, "Diagnosa COVID-19 Chest X-Ray Menggunakan Arsitektur Inception Resnet," vol. 2, no. 1, pp. 57–66, 2021.
- [17] E. Andreas and W. Widhiarso, "Klasifikasi Penyakit Mata Katarak Menggunakan Convolutional Neural Network Dengan Arsitektur Inception V3," *MDP Student Conf.*, vol. 2, no. 1, pp. 107–113, 2023, doi: 10.35957/mdp-sc.v2i1.3660.
- [18] F. Masrurroh, B. Surarso, and B. Warsito, "Perbandingan Kinerja Inception- Resnetv2, Xception, Inception-v3, dan Resnet50 pada Gambar Bentuk Wajah," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 1, pp. 11–20, 2023, doi: 10.25126/jtiik.20231014941.
- [19] L. Quaranta, F. Calefato, and F. Lanubile, "KGTorrent: A dataset of python jupyter notebooks from kaggle," *Proc. - 2021 IEEE/ACM 18th Int. Conf. Min. Softw. Repos. MSR 2021*, pp. 550–554, 2021, doi: 10.1109/MSR52588.2021.00072.
- [20] C. A. Sugianto, "Analisis Komparasi Algoritma Klasifikasi Untuk Menangani Data Tidak Seimbang Pada Data Kebakaran Hutan," *Techno.Com*, vol. 14, no. 4, pp. 336–342, 2019, [Online]. Available: <http://publikasi.dinus.ac.id/index.php/techno/article/view/992>
- [21] B. J. Baladiah and A. Ikhsani, "Canaliculitis Od + Katarak Senile Immature Os : Laporan Kasus," *J. Kesehat. Tambusai*, vol. 2, no. 3, pp. 199–203, 2021, doi: 10.31004/jkt.v2i3.2267.
- [22] T. N. Siregar and D. Juniati, "Implementasi Dimensi Fraktal Box Counting Dan K-Means dalam Klasifikasi Jenis Penyakit Mata Berdasarkan Citra Fundus Retina," *J. Ris. dan Apl. Mat.*, vol. 07, no. 02, pp. 206–223, 2023, [Online]. Available: <https://journal.unesa.ac.id/index.php/jram/article/view/27730%0Ahttps://journal.unesa.ac.id/index.php/jram/article/view/27730/10595>
- [23] J. O. Healthcare Engineering, "Retracted: On Improved 3D-CNN-Based Binary and Multiclass Classification of Alzheimer's Disease Using Neuroimaging Modalities and Data Augmentation Methods," *J. Healthc. Eng.*, vol. 2023, p. 9769464, 2023, doi: 10.1155/2023/9769464.
- [24] R. Poojary, R. Raina, and A. K. Mondal, "Effect of data-augmentation on fine-tuned cnn model performance," *IAES Int. J. Artif. Intell.*, vol. 10, no. 1, pp. 84–92, 2021, doi: 10.11591/ijai.v10.i1.pp84-92.
- [25] Luqman Hakim, Z. Sari, and H. Handhajani, "Klasifikasi Citra Pigmen Kanker Kulit Menggunakan Convolutional Neural Network," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 379–385, 2021, doi: 10.29207/resti.v5i2.3001.
- [26] R. Valentina, S. Rostianingsih, A. N. Tjondrowiguno, and J. S. Surabaya, "Pengenalan Gambar Botol Plastik dan Kaleng Minuman Menggunakan Metode Convolutional Neural Network," *J. Infra*, vol. 8, no. 1, pp. 249–254, 2020.
- [27] A. Kholik, "Klasifikasi Menggunakan Convolutional Neural Network (Cnn) Pada Tangkapan Layar Halaman Instagram," *J. Data Min. dan Sist. Inf.*, vol. 2, no. 2, p. 10, 2021, doi: 10.33365/jdmsi.v2i2.1345.
- [28] A. Akram, K. Fayakun, and H. Ramza, "Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network," *Build. Informatics, Technol. Sci.*, vol. 5, no. 2, pp. 397–406, 2023, doi: 10.47065/bits.v5i2.4063.
- [29] Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, and Galih Wasis Wicaksono, "Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1216–1222, 2021, doi: 10.29207/resti.v5i6.3607.

- [30] Y. B. E. Purba, N. F. Saragih, A. P. Silalahi, and ..., "Perancangan Alat Pendeteksi Kematangan Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode Convolutional Neural Network (CNN)," *J. Ilm. Tek. ...*, vol. 2, no. 1, pp. 13–21, 2022, [Online]. Available: <https://ojs.fikom-methodist.net/index.php/methotika/article/view/43%0Ahttps://ojs.fikom-methodist.net/index.php/methotika/article/download/43/39>
- [31] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J. Sains Komput. Inform.*, vol. 5, no. 2, pp. 697–711, 2021.
- [32] N. D. Miranda, L. Novamizanti, S. Rizal, F. T. Elektro, and U. Telkom, "CONVOLUTIONAL NEURAL NETWORK PADA KLASIFIKASI SIDIK JARI MENGGUNAKAN RESNET-50 CLASSIFICATION OF FINGERPRINT PATTERN USING CONVOLUTIONAL NEURAL NETWORK IN CLAHE IMAGE," vol. 1, no. 2, pp. 61–68, 2020.