

IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK DENGAN PRE-TRAINED MODEL MOBILENETV2 UNTUK DETEKSI KOLESTEROL

¹⁾ Indah Sri Lestari, ²⁾ Jumadi, ³⁾ Nur Lukman

^{1,2,3)} Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Gunung Djati Bandung
^{1,2,3)} Jl. A.H. Nasution No.105, Kota Bandung - Jawa Barat - Indonesia

E-mail : indahsri11209@gmail.com, jumadi@uinsgd.ac.id, n.lukman@uinsgd.ac.id

ABSTRAK

Salah satu risiko utama penyakit jantung dan stroke adalah kolesterol. Kolesterol adalah sejenis lemak yang diproduksi terutama oleh hati dan diserap dalam jumlah kecil dari makanan. Kadar kolesterol ideal dalam tubuh manusia sebaiknya kurang dari 200 mg/dl. Salah satu cara untuk memeriksa kadar kolesterol adalah dengan tes gula darah yang mengharuskan pasien berpuasa selama 10 hingga 12 jam. Mengingat bahaya yang ditimbulkan oleh kadar kolesterol tinggi, maka perlu adanya metode skrining dini yang praktis untuk mendeteksi kadar kolesterol tinggi dalam tubuh manusia. Iridologi adalah analisis iris mata untuk mendeteksi kondisi kesehatan dan menunjukkan hubungan pola iris mata dengan kadar kolesterol. Iris mata mempunyai keistimewaan tersendiri karena mampu mencatat keadaan seluruh organ, struktur tubuh, dan keadaan psikologis. Oleh karena itu, iridologi dapat menjadi alternatif analisis medis. Penelitian ini mengusulkan penggunaan algoritma jaringan saraf konvolusional menggunakan model MobileNetV2 yang telah dilatih sebelumnya. Dataset citra iris mata yang digunakan terdiri dari 200 citra yang diklasifikasikan menjadi dua kelas yaitu citra mata normal dan mata kolesterol. Hasil penelitian menunjukkan bahwa model yang diusulkan mampu mencapai akurasi sebesar 95%. Hasil ini menunjukkan bahwa model ini memiliki potensi besar sebagai alat yang praktis dan hemat biaya untuk mendeteksi kolesterol. Diperlukan penelitian lebih lanjut dengan kumpulan data yang lebih besar untuk meningkatkan akurasi dan validitas.

Kata Kunci: Kolesterol, Iridologi, Convolutional Neural Network, Transfer Learning, MobileNetV2.

ABSTRACT

High cholesterol levels are a major risk factor for heart disease and stroke. Cholesterol is a type of fat that is primarily produced by the liver and absorbed in small amounts from food. The ideal cholesterol level in the human body is less than 200 mg/dl. One way to check cholesterol levels is with a blood test that requires patients to fast for 10 to 12 hours. Given the dangers of high cholesterol levels, there is a need for a practical early screening method to detect high cholesterol levels in the human body. Iridology is the analysis of the iris to detect health conditions and shows the relationship between iris patterns and cholesterol levels. The iris has a unique ability to record the state of all organs, body structure, and psychological state. Therefore, iridology can be an alternative medical analysis. This study proposes the use of a convolutional neural network algorithm using the pre-trained MobileNetV2 model. The iris image dataset used consists of 200 images classified into two classes, namely normal eye images and cholesterol-prone eye images. The results of the study show that the proposed model is able to achieve an F1 score accuracy of 89.6%. This result shows that this model has great potential as a practical and cost-effective tool for detecting cholesterol. Further research is needed with a larger dataset to improve accuracy and validity.

Keyword: Cholesterol, Iridology, Convolutional Neural Network, Transfer Learning, MobileNetV2.

PENDAHULUAN

Berdasarkan data dari Kementerian Kesehatan Republik Indonesia pada 2022 menunjukkan bahwa penyakit jantung koroner menyumbang sebanyak 14,4% dari total kasus kematian di Indonesia [1]. Salah satu faktor utama yang berkontribusi terhadap penyakit jantung koroner adalah tingginya kadar kolesterol dalam darah [2]. Dalam dunia medis, kolesterol

digambarkan sebagai kumpulan senyawa lemak kompleks yang diproduksi oleh tubuh dan berasal dari zat makanan. Jika kadar kolesterol dalam darah berlebihan, dapat menyebabkan penyakit [3]. Untuk mengetahui kadar kolesterol, umumnya tenaga medis melakukan pemeriksaan gula darah. Saat akan melakukan pemeriksaan kolesterol, biasanya pasien diharuskan untuk berpuasa selama 12-16 jam. Kadar kolesterol normal rata-rata dalam darah

umumnya 200mg/dL. Apabila kadar kolesterol kurang dari 200mg/dL, maka masih di batas normal [4].

Menurut survey yang dilakukan oleh salah satu platform informasi kesehatan, HonestDocs, menyatakan bahwa sekitar 70% dari 9.612 responden mengaku tidak pernah melakukan tes gula darah sama sekali untuk memeriksa kadar kolestrol. Alasan orang Indonesia tidak melakukan tes darah beragam, 51% mengatakan tidak mempunyai waktu untuk pergi ke layanan kesehatan, 24% mengaku takut, 15% merasa belum memerlukan tes darah, dan 7% lainnya beralasan karena keterbatasan biaya [5].

Ada cara alternatif untuk mendeteksi kadar kolestrol yaitu dengan memanfaatkan anggota tubuh tertentu yang lebih mudah diakses seperti iris mata [6]. Iris merupakan bagian penting dari mata yang mengatur jumlah cahaya yang masuk. Cabang ilmu yang mempelajari pola dan susunan serat pada iris mata dikenal sebagai iridologi. Ahli iridologi telah lama mempelajari bentuk dan pola iris untuk mendeteksi indikasi masalah kesehatan. Penelitian menunjukkan bahwa bentuk dan pola iris dapat memberikan petunjuk diagnosa tentang kondisi kesehatan. Salah satu kondisi yang dapat dideteksi melalui iris adalah tingginya kadar kolesterol yang ditandai dengan perubahan pola iris yang disebut *Corneal Arcus* (CA) [7].

Menurut penelitian yang dilakukan oleh RA Ramlee et al. (2020), *Corneal Arcus* (CA) merupakan lapisan berwarna putih atau abu-abu muda yang terbentuk di sekitar lingkaran mata dan membentuk cincin. CA terjadi karena penimbunan sisa zat lemak (lipid). Kehadiran CA ini mengindikasikan bahwa seseorang mungkin memiliki kadar kolesterol atau trigliserida yang tinggi [8].

Melihat cincin kolesterol menggunakan mata secara langsung tidaklah mudah, karena setiap

individu memiliki struktur iris yang berbeda. Dengan berkembangnya teknologi, banyak peneliti yang membuat inovasi untuk mendeteksi kolestrol melalui iris [9].

Bersamaan dengan perkembangan zaman, kemajuan signifikan terjadi dalam bidang AI, khususnya dalam cabang utamanya, yaitu machine learning (ML). Dalam dunia kesehatan, banyak sistem yang memanfaatkan ML [10]. Evolusi ML mencapai tahap yang lebih mutakhir dengan munculnya deep learning (DL), sebuah pendekatan yang lebih kompleks namun jauh lebih canggih. Keunggulan utamanya terletak pada tingkat akurasi yang lebih tinggi, terutama dalam mengenali aktivitas atau objek. Salah satu penerapan DL adalah pada pemrosesan citra. Dalam dunia kesehatan, pemrosesan citra medis menggunakan deep learning menjadi menarik untuk diterapkan pada aplikasi yang memerlukan performa tinggi untuk penghematan waktu. Selain itu, pemrosesan citra medis menggunakan *deep learning* menjadi menarik untuk diterapkan pada aplikasi yang memerlukan performa tinggi untuk penghematan waktu [11].

Telah banyak penelitian untuk mendeteksi keberadaan *Arcus Senilis* melalui citra iris mata yang memanfaatkan ML dan turunannya seperti penelitian yang dilakukan oleh R.A Ramlee, et.al (2020) menggunakan metode *Gray Level Co-Occurance Matrix* (GLCM) dengan tingkat akurasi 98,4% [12]. Selanjutnya penelitian L.B Rachman (2020) yang menerapkan metode *K-Nearest Neighbor* (KNN) dan ekstraksi Ciri *Moment Invariant* menghasilkan akurasi sebesar 84,84% [13]. Serta, penelitian yang dilakukan oleh S. Arum Husni (2021) yang menerapkan metode *Convolutional Neural Network* (CNN) menghasilkan akurasi tinggi yaitu 97,45% [14].

Berdasarkan penelitian diatas, implementasi algoritma CNN untuk klasifikasi kolestrol

melalui pemrosesan citra iris mata memperoleh nilai akurasi yang tinggi. Penelitian telah menunjukkan bahwa penggunaan pemrosesan citra iris mata untuk klasifikasi kolesterol dapat menghasilkan tingkat akurasi yang tinggi. Hal ini didukung oleh hasil penelitian yang menyatakan bahwa Convolutional Neural Network (CNN) dianggap sebagai model terbaik untuk memecahkan masalah deteksi dan klasifikasi objek, termasuk dalam bidang citra medis [15].

Pendekatan CNN dengan melatih data dari awal sangat memakan waktu dan komputasi yang tinggi, serta pendekatan lainnya melalui *transfer learning*, sebagai jawaban untuk mengatasi kekurangan jika melatih model CNN dari awal. *Transfer learning* memungkinkan untuk menggunakan model yang sudah dilatih sebelumnya menggunakan model CNN yang telah dilatih (*pre-trained model*). Hal ini dapat menghemat banyak waktu dan sumber daya komputasi, karena tidak perlu melatih model dari awal [16].

Diantara arsitektur CNN yang telah digunakan untuk penelitian di bidang medis, khususnya deteksi kolesterol melalui iris mata adalah ResNet, Inception, VGG dengan masing-masing akurasi 88%, 88%, dan 83% [17].

Salah satu contoh *pre-trained model* yang sangat populer adalah MobileNetV2, yang dilatih menggunakan dataset ImageNet sebanyak 1,4 juta gambar. MobileNetV2 memiliki ukuran kecil dan komputasi rendah, berbeda dengan pre-trained lainnya yang cukup berat seperti ResNet dan VGG-16. Keunggulan lainnya adalah penggunaan konvolusi *depthwise separable* yang efisien secara komputasi, mengurangi jumlah parameter dalam model [18]. MobileNetV2 memberikan hasil yang lebih baik dibanding arsitektur lainnya dalam mendeteksi COVID-19 [19].

Berdasarkan latar belakang diatas, peneliti tertarik untuk melakukan implementasi

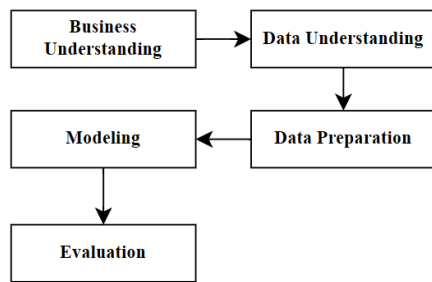
algoritma CNN dengan menggunakan arsitektur model MobileNetV2. Pendekatan ini belum banyak dieksplorasi, sehingga dapat memberikan kontribusi signifikan terhadap literatur dalam bidang ini dan membuka peluang untuk inovasi lebih lanjut dalam diagnostik medis. Selain itu, dengan teknologi yang lebih mudah diakses, deteksi kolesterol bisa dilakukan lebih sering dan di berbagai lokasi, membantu masyarakat yang memiliki keterbatasan waktu dan biaya untuk mendapatkan pemeriksaan kesehatan yang lebih teratur. Pendekatan transfer learning dengan MobileNetV2 juga diharapkan dapat menghemat waktu dan sumber daya komputasi, menghasilkan hasil yang lebih cepat tanpa perlu melatih model dari awal.

Namun, penelitian ini juga memiliki beberapa limitasi seperti model ini tidak dapat digunakan sebagai diagnosa medis yang akurat, melainkan hanya memberikan prediksi awal dan memerlukan validasi klinis lebih lanjut untuk mendapatkan hasil yang lebih jelas dan akurat mengenai kolesterol.

Harapannya, metode yang diusulkan dalam penelitian ini dapat menghasilkan hasil yang lebih unggul, menjadi landasan untuk pembangunan sistem yang dapat mendeteksi penyakit kolesterol secara dini melalui pengolahan citra iris mata.

METODE

Metodologi yang digunakan dalam penelitian ini adalah *Cross Industry Standard Process for Data Mining* (CRISP-DM) yang merupakan tandar industri untuk pengembangan data mining [20]. Kerangka penelitian yang ditunjukkan pada Gambar 1 adalah sebagai berikut.



Gambar 1. Kerangka Kerja Penelitian

Business Understanding

Fokus utama pada business understanding adalah memahami permasalahan bisnis serta tujuan yang ingin dicapai melalui sistem yang akan dibuat. Penelitian ini bertujuan untuk mengidentifikasi keberadaan *Corneal Arcus* dalam iris mata yang merupakan tanda adanya kolesterol. Untuk mencapai tujuan ini, diperlukan data yang mewakili gambar iris mata dengan *Corneal Arcus* dan tanpa *Corneal Arcus* (normal). Data yang telah diperoleh kemudian digunakan untuk membangun sebuah model yang mampu melakukan klasifikasi iris mata yang berkolesterol dan tidak. Model ini menggunakan algoritma CNN dengan arsitektur MobileNetV2. Model yang sudah dibangun akan disimpan dalam bentuk .h5.

Data Understanding

Dalam penelitian ini, dataset terbagi menjadi dua kelompok, yaitu gambar iris mata yang normal dan gambar iris mata yang mengandung *Corneal Arcus*. Untuk dataset iris mata normal diambil dari sumber database UBIRIS V2 [21]. Sementara itu, untuk mata yang memiliki gangguan didapat dengan cara *scrapping* dari website medis yang berkaitan dengan *Corneal Arcus*.



Data Preparation

1. Data Labeling

Sebelum memulai proses pelatihan model, dibutuhkan pemberian label kelas pada setiap gambar untuk mengumpulkan objek yang akan

dideteksi. Pada penelitian ini terdapat 2 kelas diantaranya kolesterol dan normal. Untuk melakukan pelabelan pada setiap kelas, akan dibuat 2 folder dengan nama masing-masing kelas tersebut. Gambar-gambar iris mata yang terdapat *Corneal Arcus* akan dimasukkan ke dalam folder kolesterol, sedangkan iris mata normal akan masuk ke folder normal. Jumlah masing-masing kategori adalah 100 gambar, sehingga total 200 gambar. Contoh data labeling ditunjukkan pada tabel 1 berikut.

Tabel 1. Tabel Data Labeling

No	Gambar	Label
1		Normal
2		Kolesterol

2. Data Pre-processing

Setelah dataset disiapkan, langkah selanjutnya adalah melakukan penyesuaian ukuran gambar input menjadi 224x224 piksel, yang merupakan ukuran standar yang digunakan oleh pre-trained model MobileNetV2. Gambar-gambar yang akan dimasukkan ke dalam MobileNetV2 biasanya memiliki nilai piksel dalam rentang [0, 255], namun MobileNetV2 hanya menerima gambar dengan rentang nilai piksel [-1, 1]. Oleh karena itu, dilakukan normalisasi (rescale) dengan cara mengurangi nilai piksel sebesar 127.5, dan kemudian membaginya dengan 127.5. Selanjutnya, dataset gambar yang terdiri dari 200 gambar dibagi menjadi tiga set. Pembagian dataset ini dijelaskan dalam Tabel 2.

Tabel 2. Splitting Data

Kategori data	jumlah	Persentasi
Data train	128	64%
Data validation	32	16%
Data test	40	20%
Total	200	100%

Modeling

Setelah melakukan pra-pemrosesan data, langkah berikutnya adalah membangun model menggunakan pre-trained model MobileNetV2. MobileNet, yang dikembangkan oleh Google, memiliki lapisan khusus yang dikenal sebagai depthwise separable convolution. Lapisan ini terdiri dari depthwise convolution dan pointwise convolution, yang bertujuan untuk mengurangi komputasi dengan mengurangi jumlah parameter, sehingga menghasilkan model yang lebih singkat. Berikut adalah arsitektur MobileNetV2 yang dijelaskan dalam Tabel 3.

Tabel 3. Arsitektur MobileNetV2

input	operator	t	c	n	s
224 ² x 3	Conv2d	-	32	1	2
112 ² x 32	bottleneck	1	16	1	1
112 ² x 16	bottleneck	6	24	2	2
56 ² x 24	bottleneck	6	32	3	2
28 ² x 32	bottleneck	6	64	4	2
14 ² x 64	bottleneck	6	96	3	1
14 ² x 96	bottleneck	6	160	3	2
7 ² x 160	bottleneck	6	320	1	1
7 ² x 320	Conv2d 1x1	-	1280	1	1
7 ² x 1280	Avgpool 7x7	-	-	1	-
1x1x1280	Conv2d 1x1	-	k	-	-

Keterangan:

n = diulang sebanyak n kali

c = ukuran output channels

s = block stride yang digunakan

t = expansion factor (expand jumlah channel)

Model MobileNetV2 menggunakan bentuk input berukuran 224x224 piksel dengan 3 channel warna RGB. Dalam model ini, terdapat beberapa jenis konvolusi dengan jumlah filter yang berbeda-beda.

Proses pertama dalam arsitektur ini adalah konvolusi 2D dengan 32 filter. Konvolusi ini bertujuan untuk menangkap fitur dasar dari gambar, seperti tepi dan tekstur. Filter dengan ukuran tertentu akan melintasi gambar, menghasilkan feature map yang mendeteksi keberadaan fitur dasar ini.

Setelah konvolusi awal, model ini dilengkapi dengan serangkaian lapisan bottleneck. Bottleneck ini terdiri dari tujuh set konvolusi dengan jumlah filter berturut-turut sebanyak 16, 24, 32, 64, 96, 160, dan 320. Fungsi utama dari bottleneck adalah untuk memperkaya representasi fitur dengan menambah kedalaman jaringan tanpa menambah beban komputasi secara signifikan.

etiap lapisan bottleneck menggunakan teknik depthwise separable convolution untuk mengurangi jumlah parameter dan komputasi tanpa mengorbankan kinerja. Teknik ini memisahkan proses konvolusi menjadi dua tahap: depthwise convolution dan pointwise convolution. Depthwise convolution melakukan konvolusi pada setiap channel input secara terpisah, sementara pointwise convolution (konvolusi 1x1) menggabungkan output dari depthwise convolution. Kombinasi ini memungkinkan pengurangan signifikan dalam jumlah komputasi yang diperlukan dibandingkan dengan konvolusi standar.

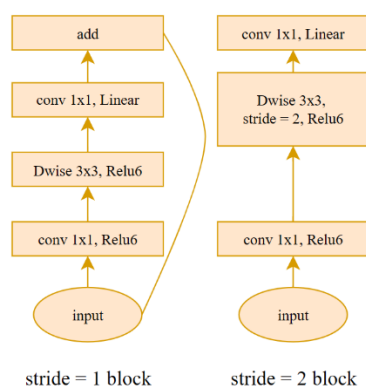
Setelah melalui lapisan bottleneck, model ini menggunakan konvolusi 1x1 dengan 1280 filter. Konvolusi ini berfungsi untuk mengubah dimensi fitur yang dihasilkan dari bottleneck menjadi lebih sesuai untuk tahap-tahap selanjutnya. Konvolusi 1x1 membantu dalam

mengkombinasikan informasi dari berbagai channel fitur tanpa mengubah resolusi spasialnya.

Selanjutnya, dilakukan average pooling dengan ukuran kernel 7×7 . Average pooling ini mengurangi dimensi spasial dari feature map dengan menghitung rata-rata nilai dalam setiap patch 7×7 . Proses ini membantu mengurangi jumlah parameter yang harus diproses pada tahap selanjutnya, sekaligus merangkum informasi fitur yang lebih luas.

Tahap terakhir dari arsitektur ini adalah konvolusi 1×1 . Konvolusi ini menggabungkan fitur-fitur yang telah diringkas oleh average pooling dan mempersiapkannya untuk klasifikasi akhir. Dalam implementasi yang umum, hasil dari konvolusi ini sering kali dilanjutkan dengan lapisan fully connected dan softmax untuk menghasilkan probabilitas klasifikasi.

Setiap proses dalam bottleneck melibatkan beberapa langkah spesifik yang diilustrasikan pada gambar 2. Gambar tersebut menunjukkan bagaimana input awal diproses melalui konvolusi depthwise, diikuti oleh konvolusi pointwise, dan akhirnya menghasilkan output yang lebih kompleks dan terperinci. Bottleneck ini berperan penting dalam meningkatkan kedalaman dan representasi fitur tanpa menambah beban komputasi yang berat.

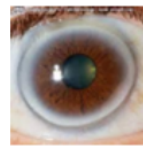


Gambar 2. Bottleneck

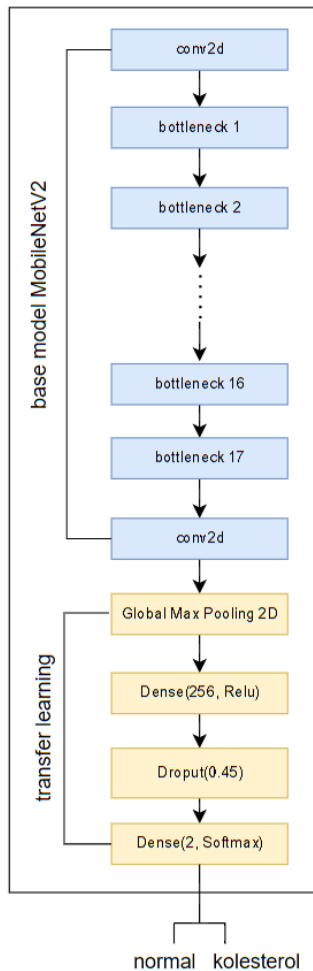
Pada penelitian ini, peneliti menggunakan *base model* dari MobileNetV2 dan melakukan *transfer learning* untuk melakukan klasifikasi dengan mengganti *classifier* nya. Arsitektur yang diterapkan disajikan pada gambar 3.

Peneliti menghapus 2 *layers* terakhir dari arsitektur MobileNetV2 dan menambah 4 *layers* baru yaitu 3 hidden layers dan 1 output layer. Penerapan pada *code* disajikan pada gambar 4.

Hidden layers terdiri dari lapisan *Global Max Pooling 2D* sebagai lapisan *flatten*, *dense layer*, dan *dropout layer*. *Dense layer* dibangun dengan 256 *neuron* dan menggunakan fungsi aktivasi ReLU. Pada *dropout layer*, sebanyak 45% *neuron* dihapus untuk mengurangi *overfitting* dan mempersingkat waktu *training*.



input
224 x 224 x 3



Gambar 3. Penerapan arsitektur Model MobileNetV2

```
inputs = Input(shape=(224, 224, 3))
x = pretrained_model(inputs, training=False) # Pass inputs through the pretrained model
x = Dense(256, activation='relu')(x)
x = Dropout(0.45)(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.45)(x)
outputs = Dense(2, activation='softmax')(x)

model = Model(inputs=inputs, outputs=outputs)

model.compile(
    optimizer=Adam(0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Gambar 4. Penerapan arsitektur Model MobileNetV2 pada code

Pada *output layer*, menggunakan fungsi aktivasi *softmax* dan 2 neuron untuk klasifikasi dua kelas. Untuk meningkatkan kinerja model *deep learning*, diterapkan *optimizer Adam* yang efisien secara komputasional. Proses pelatihan model dilakukan dengan menggunakan metode `model.fit`, yang mencakup beberapa parameter penting. Pertama, dataset pelatihan `train_images` digunakan untuk melatih model. Selanjutnya, dataset validasi `val_images` digunakan untuk mengevaluasi model selama pelatihan. Pelatihan dilakukan selama 100 epoch `epochs=100` seperti pada gambar 5.

```
history = model.fit(
    train_images,
    steps_per_epoch=len(train_images),
    validation_data=val_images,
    validation_steps=len(val_images),
    epochs=100,
    callbacks=[
        early_stopping,
        create_tensorboard_callback("training_logs",
            "iris_classification"),
        checkpoint_callback,
    ]
)
```

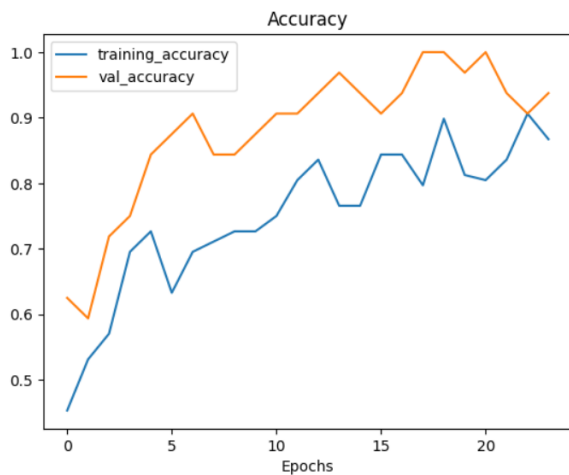
Gambar 5. penerapan model pada code. Beberapa callback juga digunakan selama pelatihan untuk meningkatkan kinerja dan efisiensi. Salah satunya adalah callback `early_stopping` yang menghentikan pelatihan lebih awal jika tidak ada peningkatan.

HASIL

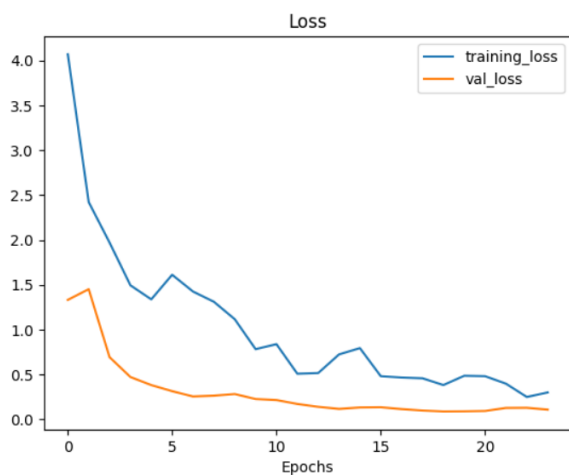
Gambar 6. merupakan proses pelatihan data yang berhenti pada epoch ke-24. Proses ini berhenti karena adanya `early_stopping`.

```
4/4 [====] - 7s 2s/step - loss: 0.7817 - accuracy: 0.7266 - val_loss: 0.2256 - val_accuracy: 0.9759
Epoch 11/100
4/4 [====] - 7s 2s/step - loss: 0.8374 - accuracy: 0.7500 - val_loss: 0.2143 - val_accuracy: 0.9062
Epoch 12/100
4/4 [====] - 10s 3s/step - loss: 0.5077 - accuracy: 0.8047 - val_loss: 0.1705 - val_accuracy: 0.9062
Epoch 13/100
4/4 [====] - 8s 2s/step - loss: 0.5149 - accuracy: 0.8359 - val_loss: 0.1384 - val_accuracy: 0.9375
Epoch 14/100
4/4 [====] - 8s 2s/step - loss: 0.7232 - accuracy: 0.7656 - val_loss: 0.1156 - val_accuracy: 0.9688
Epoch 15/100
4/4 [====] - 9s 2s/step - loss: 0.7934 - accuracy: 0.7656 - val_loss: 0.1315 - val_accuracy: 0.9375
Epoch 16/100
4/4 [====] - 8s 2s/step - loss: 0.4801 - accuracy: 0.8438 - val_loss: 0.1337 - val_accuracy: 0.9062
Epoch 17/100
4/4 [====] - 7s 2s/step - loss: 0.4653 - accuracy: 0.8438 - val_loss: 0.1133 - val_accuracy: 0.9375
Epoch 18/100
4/4 [====] - 9s 2s/step - loss: 0.4569 - accuracy: 0.7969 - val_loss: 0.0972 - val_accuracy: 1.0000
Epoch 19/100
4/4 [====] - 7s 2s/step - loss: 0.3814 - accuracy: 0.8984 - val_loss: 0.0868 - val_accuracy: 1.0000
Epoch 20/100
4/4 [====] - 9s 2s/step - loss: 0.4849 - accuracy: 0.8125 - val_loss: 0.0883 - val_accuracy: 0.9688
Epoch 21/100
4/4 [====] - 9s 2s/step - loss: 0.4803 - accuracy: 0.8047 - val_loss: 0.0923 - val_accuracy: 1.0000
Epoch 22/100
4/4 [====] - 7s 2s/step - loss: 0.3966 - accuracy: 0.8359 - val_loss: 0.1264 - val_accuracy: 0.9375
Epoch 23/100
4/4 [====] - 8s 2s/step - loss: 0.2484 - accuracy: 0.9062 - val_loss: 0.1278 - val_accuracy: 0.9062
Epoch 24/100
4/4 [====] - 7s 2s/step - loss: 0.2989 - accuracy: 0.8672 - val_loss: 0.1074 - val_accuracy: 0.9375
```

Gambar 6. Proses pelatihan model

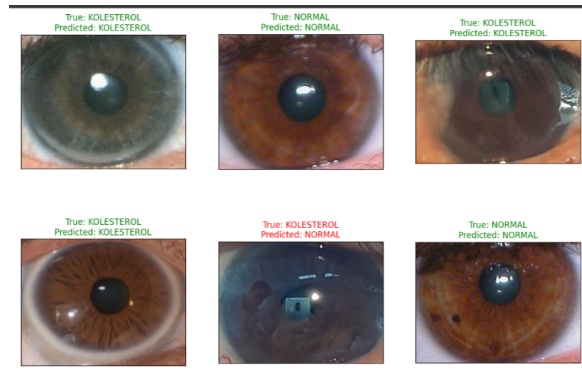


Gambar 7. Grafik akurasi train dan val



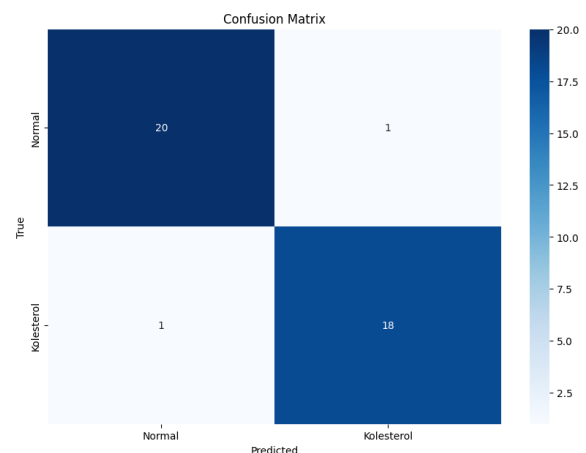
Gambar 8. Grafik loss train dan val

Pada gambar 7. dan 8. ditampilkan grafik akurasi dan loss pada saat pelatihan dan validasi. Selama proses pelatihan, model berhasil mencapai tingkat akurasi sebesar 86,72%, menunjukkan kemampuannya dalam memahami pola-pola yang signifikan dari data latihan. Pada tahap validasi, akurasi model meningkat menjadi 93,75%, menandakan kemampuannya dalam menggeneralisasi pada data yang belum pernah dilihat sebelumnya. Hasil pengujian pada data test menghasilkan akurasi sebesar 95%. Gambar 9. memvisualisasikan hasil pengujian.



Gambar 9. Hasil pengujian pada data test

Evaluasi tambahan dilakukan dengan menggunakan Confusion Matrix yang meliputi akurasi, presisi, recall, dan F1-score. Confusion Matrix merupakan tabel yang memuat empat kemungkinan kombinasi angka prediksi dan aktual, yang digunakan sebagai indikator kinerja untuk masalah klasifikasi dalam machine learning [22]. *Confusion Matrix* disajikan pada gambar 10.



Gambar 10. Confusion Matrix

Precision

Precision mengukur tingkat presisi antara data yang diminta dan hasil prediksi yang diberikan oleh model. Secara khusus, akurasi adalah rasio prediksi positif yang sebenarnya terhadap hasil prediksi positif secara keseluruhan. Dengan kata lain, akurasi mengukur seberapa banyak prediksi positif yang dibuat model yang benar-benar merupakan data positif.

$$Precision = \frac{TP}{TP + FP}$$

$$\begin{aligned}
 &= \frac{20}{20 + 1} = 0.9523 \\
 &= 0.9523 \times 100\% \\
 &= 95,23\%
 \end{aligned}$$

Recall

Recall mengukur keberhasilan model dalam mengambil informasi. Oleh karena itu, recall adalah rasio prediksi positif yang sebenarnya dibandingkan dengan total data positif yang sebenarnya.

$$\begin{aligned}
 \text{Recall} &= \frac{TP}{TP + FN} \\
 &= \frac{20}{20 + 1} = 0.9523 \\
 &= 0.9523 \times 100\% \\
 &= 95,23\%
 \end{aligned}$$

Accuracy

Akurasi menggambarkan seberapa akurat model mengklasifikasikan data. Oleh karena itu, akurasi adalah rasio prediksi yang benar (positif dan negatif) terhadap total data. Sederhananya, akurasi mencerminkan seberapa dekat nilai prediksi dengan nilai sebenarnya (sebenarnya).

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\
 \text{Accuracy} &= \frac{20 + 18}{20 + 18 + 1 + 1} = 0.95 \\
 &= 0.95 \times 100\% \\
 &= 95\%
 \end{aligned}$$

F1 Score

F1 score adalah cara untuk mengukur seberapa baik sebuah model dapat mengenali kasus

positif. Ini menggabungkan dua metrik utama yaitu precision (ketepatan) dan recall (kemampuan model untuk menemukan semua kasus positif).

$$\begin{aligned}
 \text{F1 Score} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \\
 &= 2 \times \frac{0.9523 \times 0.9523}{0.9523 + 0.9523} = 0.9523 \\
 &= 0.9523 \times 100\% \\
 &= 95,23\%
 \end{aligned}$$

KESIMPULAN

Dari hasil evaluasi model, dapat disimpulkan bahwa model yang dilatih menggunakan MobileNetV2 dengan teknik transfer learning menunjukkan kinerja yang sangat baik. Dengan akurasi pelatihan sebesar 86,72% dan akurasi validasi sebesar 93,75%, model ini mampu mengenali data baru dengan tingkat keberhasilan yang tinggi. Saat diuji dengan dataset uji, model mencapai akurasi 95%, menandakan performa yang cukup baik. Kemudian, dilakukan analisis lebih lanjut menggunakan confusion matrix. Precision sebesar 0,9523, recall 0,9523, accuracy 0,95 dan F1-score 0,9523.

Model ini menunjukkan potensi yang baik dalam klasifikasi iris mata normal dan kolesterol, sehingga layak dipertimbangkan untuk aplikasi medis. Namun, penelitian ini memiliki beberapa keterbatasan. Kualitas dan kuantitas dataset sangat mempengaruhi hasil, dan validasi klinis lebih lanjut diperlukan untuk memastikan keandalan dalam praktik medis. Struktur iris yang kompleks dan beragam antar individu juga dapat menyulitkan deteksi dan klasifikasi. Penelitian lebih lanjut dengan dataset yang lebih besar diperlukan untuk meningkatkan akurasi dan validitas.

DAFTAR PUSTAKA

- [1] “Penyakit Jantung Penyebab Utama Kematian, Kemenkes Perkuat Layanan Primer – Sehat Negeriku.” Accessed: Dec. 04, 2023. [Online]. Available: <https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20220929/0541166/penyakit-jantung-penyebab-utama-kematian-kemenkes-perkuat-layanan-primer/>
- [2] T. Kocejko, J. Ruminski, M. Mazur-Milecka, M. Romanowska-Kocejko, K. Chlebus, And K. H. Jo, “Using Convolutional Neural Networks For Corneal Arcus Detection Towards Familial Hypercholesterolemia Screening,” *Journal Of King Saud University - Computer And Information Sciences*, Vol. 34, No. 9, Pp. 7225–7235, Oct. 2022, Doi: 10.1016/j.jksuci.2021.09.001.
- [3] S. N. Andana, L. Novamizanti, And I. N. Apraz Ramatryana, “Measurement Of Cholesterol Conditions Of Eye Image Using Fuzzy Local Binary Pattern (Flbp) And Linear Regression,” *Proceedings - 2019 Ieee International Conference On Signals And Systems, Icsigsys 2019*, Pp. 79–84, Jul. 2019, Doi: 10.1109/Icsigsys.2019.8811071.
- [4] I. Gusti Ayu Sri Ekayanti, “Analisis Kadar Kolesterol Total Dalam Darah Pasien Dengan Diagnosis Penyakit Kardiovaskuler,” *International Journal Of Applied Chemistry Research* |, Vol. 1, No. 1, Pp. 2541–7207, 2019, Doi: 10.23887/ijacr-Undiksha.
- [5] “Banyak Orang Indonesia Tak Pernah Cek Darah Padahal Penting - Antara News.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.antaranews.com/berita/1114794/banyak-orang-indonesia-tak-pernah-cek-darah-padahal-penting>
- [6] A. Munjal And E. J. Kaufman, “Arcus Senilis,” *Statpearls*, Jul. 2023, Accessed: Jun. 11, 2024. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/nbk554370/>
- [7] M. Daniel, J. Raharjo, And K. Usman, “Iris-Based Image Processing For Cholesterol Level Detection Using Gray Level Co-Occurrence Matrix And Support Vector Machine,” *Engineering Journal*, Vol. 24, No. 5, Pp. 135–144, Sep. 2020, Doi: 10.4186/Ej.2020.24.5.135.
- [8] R. A. Ramlee, S. K. Subramaniam, S. B. Yaakob, A. S. F. Rahman, And N. M. Saad, “Corneal Arcus Classification For Hyperlipidemia Detection Using Gray Level Co-Occurrence Matrix Features,” In *Journal Of Physics: Conference Series*, Institute Of Physics Publishing, Jan. 2020. Doi: 10.1088/1742-6596/1432/1/012084.
- [9] A. Fadlil, W. S. Aji, And A. S. Nugroho, “Sistem Monitoring Kolesterol Melalui Iris Mata Dengan Metode Pengolahan Citra,” *Jurnal Rekayasa ElektriKa*, Vol. 16, No. 1, May 2020, Doi: 10.17529/Jre.V16i1.15657.
- [10] R. A. D. Yulianto, I. Riadi, And R. Umar, “Perancangan Klasifikasi Pasien Stroke Dengan Metode K-Nearest Neighbor,” *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, Vol. 8, No. 2, Pp. 262–268, Sep. 2023, Doi: 10.36341/Rabit.V8i2.3454.
- [11] C. Castaneda *Et Al.*, “Clinical Decision Support Systems For Improving Diagnostic Accuracy And Achieving Precision Medicine,” *J Clin Bioinforma*, Vol. 5, No. 1, Dec. 2015, Doi: 10.1186/S13336-015-0019-3.
- [12] R. A. Ramlee, K. A. Aziz, S. Ranjit, And M. Esro, “Automated Detecting Arcus Senilis, Symptom For Cholesterol Presence Using Iris Recognition Algorithm,” *Journal Of Telecommunication, Electronic And Computer Engineering (Jtec)*, Vol. 3, No. 2, Pp. 29–39, 2011.
- [13] S. Arum Nurhusni And R. Ibnu Adam, “Klasifikasi Kadar Kolesterol Menggunakan Ekstraksi Ciri Moment Invariant Dan Algoritma K-Nearest Neighbor (Knn),” 2021. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/jaic>
- [14] L. B. Rachman And Basari, “Detection Of Cholesterol Levels By Analyzing Iris Patterns Using Backpropagation Neural Network,” In *Iop Conference Series: Materials Science And Engineering*, Institute Of Physics Publishing, Jul. 2020. Doi: 10.1088/1757-899x/852/1/012157.
- [15] I. F. Alam, M. Ihsan Sarita, And A. M. Sajiah, “Terakreditasi ‘Peringkat 4 (Sinta 4)’ Oleh Kemenristekdikti Implementasi Deep Learning Dengan Metode Convolutional Neural Network Untuk Identifikasi Objek Secara Real Time Berbasis Android,” Vol. 5, No. 2, Pp. 237–244, Doi: 10.5281/Zenodo.3459374.
- [16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, And L. C. Chen, “Mobilenetv2: Inverted Residuals And Linear Bottlenecks,” *Proceedings Of The Ieee Computer Society Conference On Computer Vision And Pattern Recognition*, Pp. 4510–4520, Jan. 2018, Doi: 10.1109/Cvpr.2018.00474.
- [17] T. Kocejko, J. Ruminski, M. Mazur-Milecka, M. Romanowska-Kocejko, K. Chlebus, And K. H. Jo, “Using Convolutional Neural Networks For Corneal Arcus Detection Towards Familial Hypercholesterolemia Screening,” *Journal Of King Saud University - Computer And Information Sciences*, Vol. 34, No. 9, Pp. 7225–7235, Oct. 2022, Doi: 10.1016/j.jksuci.2021.09.001.
- [18] A. G. Howard *Et Al.*, “Mobilenets: Efficient Convolutional Neural Networks For Mobile Vision Applications,” Apr. 2017, [Online].

- Available: [Http://Arxiv.Org/Abs/1704.04861](http://Arxiv.Org/Abs/1704.04861)
- [19] Y. Kaya And E. Gürsoy, “A Mobilenet-Based Cnn Model With A Novel Fine-Tuning Mechanism For Covid-19 Infection Detection,” *Soft Comput*, Vol. 27, No. 9, Pp. 5521–5535, May 2023, Doi: 10.1007/S00500-022-07798-Y.
- [20] T. Iman Hermanto, Y. Muhyidin, T. Informatika, And S. Wastukencana Purwakarta, “Analisis Data Sebaran Bandwidth Menggunakan Algoritma Dbscan Untuk Menentukan Tingkat Kebutuhan Bandwidth Di Kabupaten Purwakarta,” *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, Vol. 5, No. 2, Pp. 130–137, Jul. 2020, Doi: 10.36341/Rabit.V5i2.1388.
- [21] “Ubiris.V2 Dataset | Papers With Code.” Accessed: Jun. 11, 2024. [Online]. Available: [Https://Paperswithcode.Com/Dataset/Ubiris-V2](https://Paperswithcode.Com/Dataset/Ubiris-V2)
- [22] D. Safitri *Et Al.*, “Analisis Penggunaan Algoritma Klasifikasi Dalam Prediksi Kelulusan Menggunakan Orange Data Mining,” *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, Vol. 8, No. 1, Pp. 75–81, Jan. 2023, Doi: 10.36341/Rabit.V8i1.3009.