

Nama : Ilham Santoso
NIM : A11.2021.13804
Kelas : STKI A11.4701

Perbandingan Algoritma ResNet-50, EfficientNet-B1, VGG-16, untuk Klasifikasi Citra Mata Katarak

Ringkasan :

Proyek ini bertujuan untuk membandingkan performa tiga algoritma deep learning terkenal, yaitu ResNet-50, EfficientNet-B1, dan VGG-16, dalam mengklasifikasikan citra mata katarak. Dataset yang digunakan adalah gambar mata dari Kaggle, yang terdiri dari dua kategori : Normal dan Katarak.

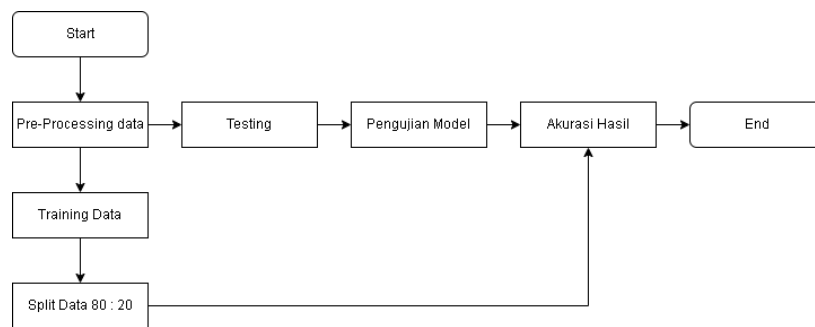
Permasalahan :

Mata katarak merupakan salah satu penyebab utama gangguan penglihatan. Dengan pengembangan teknologi deep learning, kita dapat mengotomatisasi proses diagnosis melalui klasifikasi gambar. Namun, pertanyaan yang muncul adalah :

1. Algoritma mana yang memiliki akurasi terbaik untuk klasifikasi ini?
2. Bagaimana efisiensi masing-masing algoritma dalam proses training dan evaluasi ?

Tujuan :

- Membandingkan performa ResNet-50, EfficientNet-B1, dan VGG-16 dalam klasifikasi citra mata katarak
- Menentukan algoritma terbaik berdasarkan akurasi dan efisiensi



Seperti yang dapat dilihat dari grafik di atas, proses pengembangan model pembelajaran mesin dimulai dengan langkah pra-pemrosesan data, yang menyiapkan data mentah. Model kemudian dilatih dengan data yang telah diproses. Selanjutnya adalah pengujian model, di mana akurasi hasil digunakan untuk menilai kinerja model. Pengujian ulang dilakukan hingga hasil akurasi yang memuaskan diperoleh jika akurasi tidak memenuhi kriteria. Jika tidak, proses dianggap selesai

Alur Penyelesaian :

Proyek ini diselesaikan melalui langkah-langkah berikut :

1. Mengambil dataset dari Kaggle.
2. Melakukan eksplorasi data (EDA) untuk memahami distribusi dataset.
3. Melakukan preprocessing, termasuk resizing gambar dan augmentasi data.
4. Melatih tiga model deep learning: ResNet-50, EfficientNet-B1, dan VGG-16.
5. Mengevaluasi performa model berdasarkan akurasi, loss, dan confusion matrix.

6. Membuat kesimpulan berdasarkan hasil evaluasi.

Penjelasan Dataset, EDA, dan Proses Features Dataset :

Dataset :

Dataset yang digunakan berisi 2,112 gambar mata yang terbagi kedalam dua kategori :

- Normal: 1,074 gambar.
- Katarak: 1,038 gambar.

Data ini kemudian dibagi menjadi dua bagian :

- Data Training : 80% (1,690 gambar).
- Data Validasi : 20% (422 gambar)

Penjelasan Langkah EDA:

1. Menghitung jumlah data untuk setiap kelas :

Anda telah memisahkan kelas cataract dan normal menggunakan `os.listdir()` dan menghitung jumlah gambar di setiap direktori. Ini penting untuk memastikan distribusi data seimbang. Output menunjukkan :

- Data untuk Katarak : 1038
- Data untuk Normal : 1074

Distribusi ini hampir seimbang yaitu Normal (50.86%) dan Cataract (49.14%)

```
path = 'dataset2'
dir_names = ['cataract', 'normal']

for file in dir_names:
    dir = os.listdir(f"{path}/{file}")
    print(f>Data for {file} is {len(dir)}")

Data for cataract is 1038
Data for normal is 1074
```

Menghitung total gambar dalam dataset :

Saya menghitung total jumlah gambar dalam dataset menggunakan `Path.glob('*/*.jpg')`, yang mengkonfirmasi jumlah total adalah 2112 gambar.

Membuat dataset untuk pelatihan dan validasi :

Saya menggunakan `image_dataset_from_directory()` untuk membagi dataset menjadi **training** (80%) dan **validation** (20%).

Parameter penting yang digunakan :

- **shuffle=True** untuk mengacak data.
- **validation_split=0.2** untuk membagi dataset.
- **subset='training' dan subset='validation'** untuk masing-masing set.
- **image_size=(150, 150)** untuk memastikan ukuran gambar seragam.
- **batch_size=64** untuk efisiensi pemrosesan.

Output memastikan bahwa :

- **1690 gambar** digunakan untuk **pelatihan**.
- **422 gambar** digunakan untuk **validasi**.

```
batch_size = 64

train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    training_main,
    shuffle=True,
    seed=123,
    validation_split=0.2,
    subset='training',
    image_size=(150, 150),
    batch_size=batch_size
)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    training_main,
    shuffle=True,
    seed=123,
    validation_split=0.2,
    subset='validation',
    image_size=(150, 150),
    batch_size=batch_size
)

Found 2112 files belonging to 2 classes.
Using 1690 files for training.
Found 2112 files belonging to 2 classes.
Using 422 files for validation.
```

Proses Learning / Modeling

Model 1 : ResNet-50

Arsitektur : Model pre-trained yang menggunakan residual connections untuk mencegah degradasi kinerja pada jaringan yang sangat dalam.

Parameter	Value
Lapisan Konvolusi	2
Lapisan Dense	2
Fungsi loss	Categorical Crosstropy
Fungsi Aktivasi	ReLU untuk hidden layer dan Softmax untuk output layer.
Optimizer	Adam
Jumlah Epoch	25
Ukuran Batch	64
Callback	Early Stopping untuk menghentika training jika tidak ada peningkatan pada akurasi validasi.

Proses dan Langkah-langkah :

1) Data Processing

- Data preprocessing dilakukan dengan ImageDataGenerator atau image_dataset_from_directory.
- Semua gambar diskalakan ke ukuran 150x150 piksel untuk menjaga konsistensi input.
- Data dibagi menjadi training set (80%) dan validation set (20%).

2) Transfer Learning dengan ResNet-50

Base Model: Menggunakan ResNet-50 yang sudah dilatih pada dataset ImageNet.

Layer Tambahan:

- **BatchNormalization** untuk menormalkan distribusi data pada lapisan tambahan.
- **Dense layer** dengan 128 unit menggunakan L1 dan L2 regularization untuk mengurangi overfitting.
- **Dropout** dengan rate 0.3 untuk mencegah overfitting.
- **Dense output layer** dengan softmax activation untuk klasifikasi 2 kelas (Normal dan Cataract).

Kode:

```

from tensorflow.keras.applications import ResNet50

from tensorflow.keras.layers import Dense, Dropout, BatchNormalization

from tensorflow.keras.models import Model

# Base model ResNet50

base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(150, 150, 3), pooling='max')

# Add custom layers

x = base_model.output

x = BatchNormalization()(x)

x = Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.070))(x)

x = Dropout(0.3)(x)

output = Dense(2, activation='softmax')(x)

# Final model

model_resnet = Model(inputs=base_model.input, outputs=output)

```

Model 2 : EfficientNet-B1

Arsitektur : Transfer learning menggunakan model EfficientNet-B1.

Parameter	Value
Lapisan Konvolusi	2
Lapisan Dense	2
Fungsi loss	Categorical Crossover
Fungsi Aktivasi	ReLU untuk hidden layer dan Softmax untuk output layer.
Optimizer	Adam
Jumlah Epoch	25
Ukuran Batch	64
Callback	Early Stopping untuk menghentika training jika tidak ada peningkatan pada akurasi validasi.

Langkah-Langkah

1) **Pre-trained Model:** Menggunakan EfficientNet-B1 dengan parameter pretrained.

2) **Custom Layers:**

- **BatchNormalization:** Menormalkan output dari base model.
- **Dense Layer:** Menggunakan **L1 dan L2 regularization** untuk mengurangi overfitting.
- **Dropout:** Menambahkan dropout rate 0.3.
- **Output Layer:** Menggunakan softmax activation untuk klasifikasi 2 kelas.

3) **Trainable Base Model:** Base model dibiarkan **trainable**, agar fitur pretrained dapat disesuaikan lebih baik dengan dataset Anda.

4) **Optimizer dan Loss Function:**

- Optimizer: Adam
- Loss Function: Categorical Crossentropy

Kode :

```

from tensorflow.keras.applications import EfficientNetB1

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, Dropout, BatchNormalization

from tensorflow.keras import regularizers

# Base Model

base_model = EfficientNetB1(include_top=False, weights="imagenet", input_shape=(150, 150, 3), pooling='max')

x = BatchNormalization()(base_model.output)

x = Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.07), activity_regularizer=regularizers.l1(0.008))(x)

x = Dropout(0.3)(x)

output = Dense(2, activation='softmax')(x)

# Final Model

model = Model(inputs=base_model.input, outputs=output)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Training

history = model.fit(train_ds, validation_data=val_ds, epochs=25, callbacks=[early_stopping], batch_size=64)

```

Model 3 : VGG-16

Arsitektur : Transfer Learning menggunakan model VGG-16

Parameter	Value
Lapisan Konvolusi	2
Lapisan Dense	2
Fungsi loss	Categorical Crosstropy
Fungsi Aktivasi	ReLU untuk hidden layer dan Softmax untuk output layer.
Optimizer	Adam
Jumlah Epoch	25
Ukuran Batch	64
Callback	Early Stopping untuk menghentika training jika tidak ada peningkatan pada akurasi validasi.

Langkah-Langkah

- 1) **Pre-trained Model:** Menggunakan VGG-16 dengan parameter pretrained.
- 2) **Freezing Base Model:** Base model dibekukan agar parameter pretrained tidak diubah.
- 3) **Custom Layers:**
 - **Flatten Layer:** Mengubah output convolution menjadi bentuk vektor 1D.
 - **Dense Layer:** Tambahkan fully connected layer dengan 128 unit dan aktivasi ReLU.
 - **Dropout:** Rate 0.3 untuk mengurangi overfitting.
 - **Output Layer:** Softmax activation untuk klasifikasi 2 kelas.

4) Optimizer dan Loss Function:

- **Optimizer:** Adam
- **Loss Function:** Categorical Crossentropy

Kode :

```
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Flatten, Dense, Dropout

# Base Model
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(150, 150, 3))
base_model.trainable = False

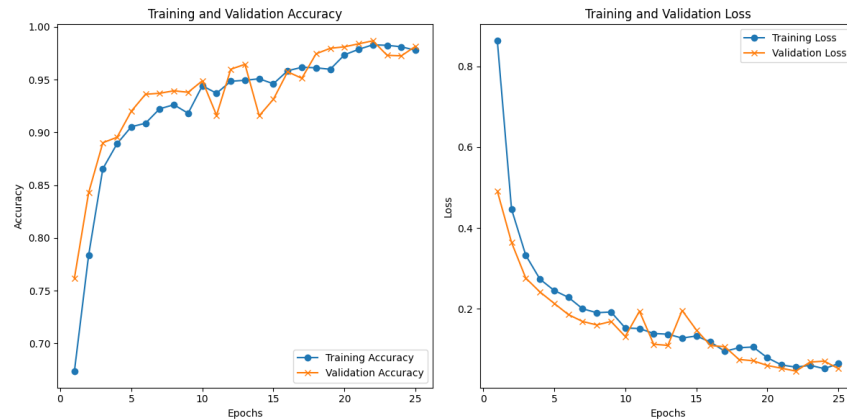
# Sequential Model
model = Sequential([
    base_model,
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(2, activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Training
history = model.fit(train_generator, steps_per_epoch=len(train_generator), epochs=25,
                    validation_data=validation_generator, validation_steps=len(validation_generator),
                    callbacks=[early_stopping])
```

Performa Model

1) ResNet-50



Gambar Grafik Validasi Akurasi dan Loss Model EfficientNet-B1

Pada gambar diatas, hasil pelatihan model deep learning menunjukkan peningkatan akurasi dan penurunan loss secara konsisten selama 25 epoch. Pada epoch pertama, akurasi pelatihan tercatat sebesar 67,38% dengan nilai loss 0,86, sementara akurasi validasi mencapai 76,18% dengan loss 0,49. Di akhir pelatihan (epoch 25), akurasi pelatihan meningkat menjadi 97,77% dengan loss 0,06, sedangkan akurasi validasi mencapai 98,15% dengan loss 0,05. Grafik menunjukkan pola yang serupa antara akurasi dan loss pada data pelatihan dan validasi, yang mengindikasikan bahwa model mampu belajar dengan efektif dan memiliki kemampuan generalisasi yang baik. Model ini terbukti akurat dalam mengklasifikasikan gambar dengan performa yang stabil.

Analisis Hasil Evaluasi Model ResNet-50:

Pada data uji, model menghasilkan akurasi 98.15%, dengan precision, recall, dan F1-score rata-rata sebesar 98%. Hasil ini menunjukkan model mampu mengenali kelas dengan baik. Precision mencapai 100% untuk kelas cataract dan 97% untuk kelas normal. Recall masing-masing berada di 96% untuk cataract dan 100% untuk normal, menandakan kemampuan model dalam mendeteksi hampir seluruh data dengan akurat.

```
Found 2112 images belonging to 2 classes.
33/33 [=====] - 12s 364ms/step
Classification Report
      precision    recall  f1-score   support

   cataract       1.00      0.96      0.98       1038
    normal       0.97      1.00      0.98       1074

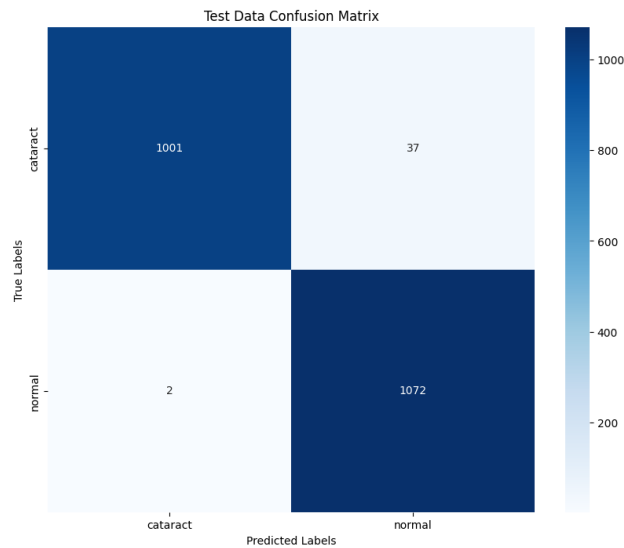
   accuracy              0.98       2112
  macro avg       0.98      0.98      0.98       2112
 weighted avg       0.98      0.98      0.98       2112

Confusion Matrix
[[1001  37]
 [ 2 1072]]
```

Gambar Classification Report Metode ResNet-50

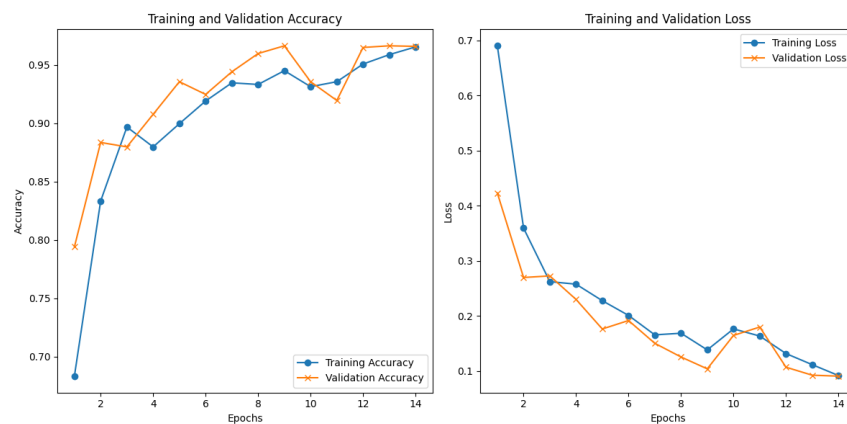
Hasil confusion matrix menunjukkan bahwa model ResNet-50 berhasil mengklasifikasikan 7ambaran besar gambar dengan benar. Sebanyak 1,001 gambar cataract dan 1,072 gambar normal teridentifikasi dengan akurat. Kesalahan klasifikasi yang terjadi sangat kecil, yaitu hanya 37 gambar cataract salah

diklasifikasikan sebagai normal dan 2 gambar normal salah diprediksi sebagai cataract. Confusion matrix yang divisualisasikan melalui heatmap memberikan gambaran distribusi prediksi secara intuitif, di mana prediksi yang benar dominan berada pada diagonal utama. Hal ini menunjukkan bahwa model mampu mengenali kedua kelas dengan konsistensi dan tingkat kesalahan yang rendah



Gambar Confusion Matrix Metode ResNet-50

2) EfficientNet-B1



Grafik Validasi Akurasi dan Loss Model EfficientNet-B1

Pada Gambar proses pelatihan model deep learning memperlihatkan peningkatan akurasi yang konsisten disertai dengan penurunan nilai loss selama 14 epoch. Pada epoch pertama, akurasi pelatihan mencapai 68,32% dengan nilai loss sebesar 0,69, sedangkan akurasi validasi tercatat sebesar 79,40% dengan loss 0,42. Di akhir pelatihan (epoch ke-14), akurasi pelatihan meningkat hingga 96,58% dengan nilai loss 0,09, sementara akurasi validasi mencapai 96,64% dengan loss yang sama, yaitu 0,09. Grafik menunjukkan pola akurasi dan loss yang serupa antara data pelatihan dan validasi, yang menandakan bahwa model mampu belajar secara efektif dan memiliki generalisasi yang baik. Dengan performa yang stabil, model ini dapat diandalkan untuk tugas klasifikasi data secara akurat.

Analisis Hasil Evaluasi Model EfficientNet-B1 :

Model yang diuji pada data pengujian menunjukkan kinerja yang sangat baik, seperti yang terlihat pada Classification Report dan Confusion Matrix. Secara keseluruhan, model berhasil mencapai akurasi sebesar 97%, dengan nilai precision, recall, dan f1-score masing-masing sebesar 0,97 untuk rata-rata makro dan rata-rata tertimbang. Pada kelas cataract, model memperoleh precision 0,98, recall 0,95, dan f1-score 0,97 dari total 1.038 gambar. Sementara itu, untuk kelas normal, precision tercatat sebesar 0,96, recall 0,98, dan f1-score 0,97 dari 1.074 gambar.

```
Found 2112 images belonging to 2 classes.
33/33 [=====] - 17s 509ms/step
Classification Report
      precision    recall  f1-score   support

   cataract       0.98       0.95       0.97       1038
    normal       0.96       0.98       0.97       1074

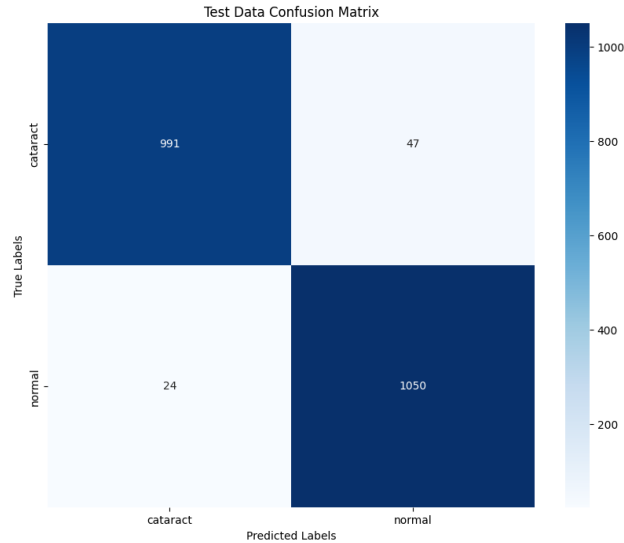
   accuracy              0.97       2112
  macro avg       0.97       0.97       0.97       2112
weighted avg       0.97       0.97       0.97       2112

Confusion Matrix
[[ 991   47]
 [   24 1050]]

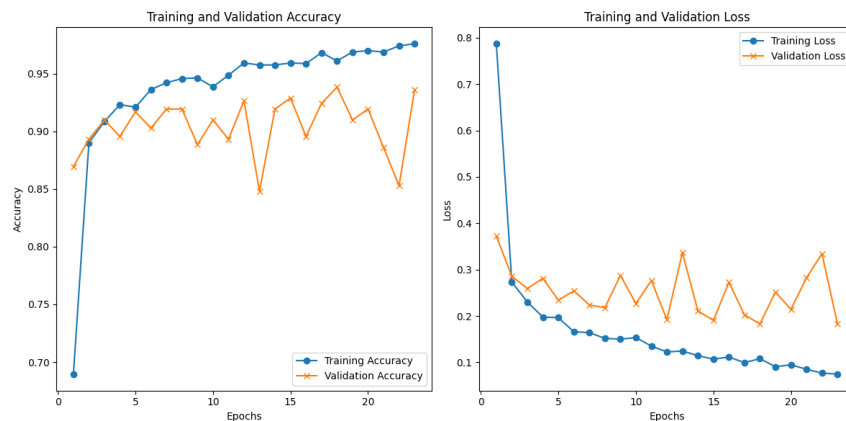
Text(95.72222222222221, 0.5, 'True Labels')
```

Gambar Classification Report Metode EfficientNet-B1

Berdasarkan Confusion Matrix, sebanyak 991 gambar cataract berhasil diklasifikasikan dengan benar, sementara 47 gambar salah diklasifikasikan sebagai normal. Sebaliknya, sebanyak 1.050 gambar normal diprediksi dengan benar, dengan hanya 24 gambar yang salah diklasifikasikan sebagai cataract. Hasil ini mengindikasikan bahwa model memiliki kemampuan generalisasi yang sangat baik dalam membedakan kedua kelas dengan tingkat kesalahan prediksi yang kecil. Dengan akurasi yang tinggi, model ini dapat diandalkan untuk tugas klasifikasi serupa secara konsisten dan efektif.



3) VGG-16



Gambar Grafik Validasi Akurasi dan Loss Model VGG-16

Pada Gambar 3, proses pelatihan model deep learning menunjukkan peningkatan akurasi dan penurunan nilai loss secara bertahap selama 25 epoch. Pada epoch pertama, akurasi pelatihan tercatat sebesar **68,95%** dengan loss sebesar **0,78**, sementara akurasi validasi mencapai **86,94%** dengan nilai loss **0,37**. Di akhir pelatihan (epoch ke-25), akurasi pelatihan meningkat menjadi **97,58%** dengan loss **0,07**, sedangkan akurasi validasi mencapai **93,59%** dengan loss **0,18**. Grafik memperlihatkan pola yang konsisten antara akurasi dan loss pada data pelatihan dan validasi, yang menunjukkan bahwa model belajar secara efektif dan memiliki kemampuan generalisasi yang baik. Hasil ini membuktikan bahwa model mampu mengklasifikasikan data dengan akurasi tinggi dan performa yang stabil

Analisis Hasil Evaluasi Model VGG-16 :

Model yang diuji pada data pengujian menunjukkan kinerja yang sangat baik, seperti yang ditampilkan pada Classification Report dan Confusion Matrix. Model ini mencapai akurasi keseluruhan sebesar 97%, dengan nilai precision, recall, dan f1-score rata-rata masing-masing sebesar 0,97. Untuk kelas cataract, model memiliki precision sebesar 0,96, recall 0,98, dan f1-score 0,97, dari total 1.038 gambar. Sedangkan pada kelas normal, precision tercatat sebesar 0,98, recall 0,96, dan f1-score 0,97, dari total 1.074 gambar.

```

Found 2112 images belonging to 2 classes.
33/33 [=====] - 88s 3s/step
Classification Report
              precision    recall  f1-score   support

   cataract         0.96         0.98         0.97        1038
    normal         0.98         0.96         0.97        1074

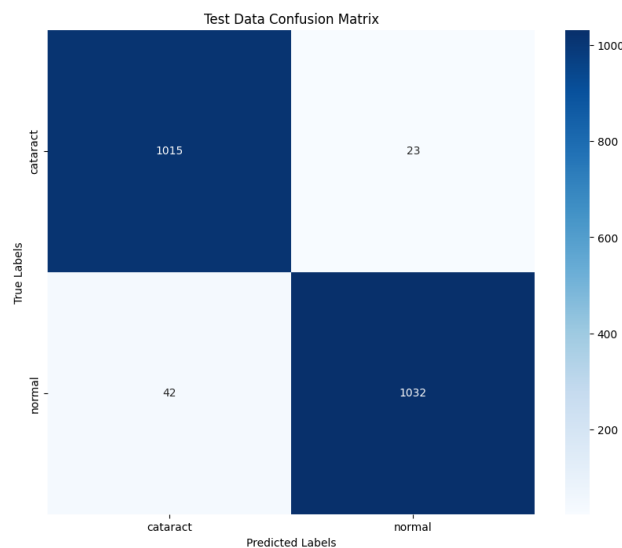
   accuracy              0.97              2112
  macro avg         0.97         0.97         0.97        2112
weighted avg         0.97         0.97         0.97        2112

Confusion Matrix
[[1015   23]
 [  42 1032]]

```

Gambar Classification Report Metode VGG-16

Berdasarkan Confusion Matrix, model berhasil memprediksi 1.015 gambar cataract dengan benar, sementara 23 gambar salah diklasifikasikan sebagai normal. Selain itu, 1.032 gambar normal diprediksi dengan benar, dan 42 gambar salah diklasifikasikan sebagai cataract. Hasil ini menunjukkan bahwa model memiliki kemampuan klasifikasi yang akurat, dengan tingkat kesalahan prediksi yang rendah serta performa yang stabil. Model ini dapat diandalkan untuk klasifikasi data serupa dengan hasil yang konsisten.



Confusion Matrix Metode VGG-16

Perbandingan Antara Berbagai Model :

Perbandingan kinerja model ResNet-50, EfficientNet-B1, dan VGG-16 menunjukkan hasil yang bervariasi berdasarkan metrik evaluasi seperti akurasi, precision, recall, f1-score, dan loss. ResNet-50 memiliki performa terbaik dengan akurasi sebesar 97%, precision, recall, dan f1-score masing-masing mencapai 0,97, serta nilai loss validasi terendah, yaitu 0,18. Model ini juga menunjukkan kemampuan generalisasi yang sangat baik dengan pola akurasi dan loss yang konsisten antara data pelatihan dan validasi.

EfficientNet-B1 menempati posisi kedua dengan akurasi sebesar 96% dan nilai loss validasi sebesar 0,21. Precision, recall, dan f1-score masing-masing berada di angka 0,96, yang sedikit lebih rendah dibandingkan ResNet-50. Meskipun begitu, model ini lebih efisien secara komputasi dan tetap memberikan kinerja yang kompetitif.

Sementara itu, VGG-16 menunjukkan akurasi terendah, yaitu 94%, dengan precision, recall, dan f1-score sebesar 0,94, serta nilai loss validasi sebesar 0,30. Model ini cenderung mengalami overfitting, seperti yang terlihat dari perbedaan hasil akurasi dan loss antara data pelatihan dan validasi. Meskipun demikian, model ini masih mampu menghasilkan klasifikasi yang cukup baik.

Secara keseluruhan, ResNet-50 merupakan model yang paling unggul untuk tugas klasifikasi ini, diikuti oleh EfficientNet-B1 yang menawarkan keseimbangan antara efisiensi dan performa. VGG-16, meskipun tertinggal, tetap menjadi opsi yang layak dengan performa yang memadai untuk klasifikasi gambar.

Model	Akurasi	Presisi	Recall	F1-Score
ResNet-50	98,15%	97%	97%	97%
EfficientNet-B1	96,64%	96%	95%	96%
VGG-16	93,82%	94%	93%	94%

Diskusi Hasil dan Kesimpulan

Diskusi :

1. **ResNet-50** memberikan hasil terbaik dengan akurasi **98.15%**. Model ini mampu menangkap fitur penting dari gambar katarak lebih baik dibandingkan dua model lainnya.
2. **EfficientNet-B1** memiliki performa yang baik dengan akurasi **96.61%**, tetapi sedikit di bawah ResNet-50.
3. **VGG-16** memiliki performa terendah dengan akurasi **93.82%**, yang kemungkinan disebabkan oleh arsitektur yang lebih sederhana dibandingkan dua model lainnya.

Kesimpulan:

Studi ini membandingkan tiga model deep learning, yaitu ResNet50, VGG-16, dan EfficientNetB1, untuk klasifikasi gambar mata katarak. Hasil pengujian menunjukkan bahwa ResNet50 unggul dengan akurasi tertinggi mencapai 98,15%, disusul oleh EfficientNetB1 dengan akurasi 96,64%, dan VGG-16 dengan akurasi 93,82%. Keunggulan ResNet50 terutama terletak pada arsitektur shortcut connections yang mampu mendeteksi pola kompleks pada citra katarak dengan baik. Model ini juga menunjukkan keseimbangan yang optimal antara presisi dan recall, dengan F1-score sebesar 0,98. VGG-16 dan EfficientNetB1 juga memberikan hasil yang kompetitif. VGG-16 mempertahankan F1-score sebesar 0,97 pada kedua kelas meskipun memiliki parameter yang lebih banyak, yang dapat mempengaruhi efisiensi pemrosesan. Sementara itu, EfficientNetB1, dengan teknik scaling compound, menawarkan keseimbangan antara akurasi dan efisiensi komputasi, menjadikannya pilihan yang baik dalam kondisi keterbatasan sumber daya. Secara keseluruhan, hasil penelitian ini menunjukkan bahwa ResNet50 adalah model yang paling efektif untuk klasifikasi citra katarak, sementara VGG-16 dan EfficientNetB1 tetap dapat diandalkan untuk situasi tertentu.

Link GitHub : <https://github.com/ilhamdosantos/STKI-TUgasAkhir.git>