

## DETEKSI MASALAH KULIT WAJAH DAN JENIS KULIT MENGUNAKAN CNN *TRANSFER LEARNING* MOBILENETV3

Honestyan Didyafarhan Atthariq<sup>1</sup>, Basuki Rahmat<sup>2</sup>, Hendra Maulana<sup>3</sup>.  
Program Studi Informatika, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur<sup>1-3</sup>

\*Korespondensi: [basukirahmat.if@upnjatim.ac.id](mailto:basukirahmat.if@upnjatim.ac.id)

### **Abstract**

*This research aims to develop an efficient and accurate skin care classification model using transfer learning and the MobileNetV3 architecture in convolutional neural networks. In Indonesia, healthcare technology plays a crucial role in the cosmetic and personal care industry, however, lack of understanding often leads consumers to make mistakes in choosing and using skincare products, which can worsen skin conditions. The MobileNetV3-Small model, pretrained beforehand, is utilized as the base model, and then refined based on specific skincare image data. Previous research results have shown that this model has the highest classification accuracy (94%) in predicting skin aging, making it the preferred choice in this study. Evaluation through confusion matrix indicates that the model performs best in the main scenario, while performance declines in other scenarios. However, scenario 4 shows performance close to the main scenario, indicating similarity in data between the two scenarios. Thus, this research is expected to assist Indonesian consumers in selecting the right skincare products and increase awareness about the importance of using products suitable for their needs and skin types.*

### **Keywords:**

Convolutional Neural Network, Transfer Learning, MobileNetV3, Classification, Skincare, Consumer Awareness

### **Abstrak**

Penelitian ini bertujuan untuk mengembangkan model klasifikasi perawatan kulit yang efisien dan akurat menggunakan pembelajaran transfer dan arsitektur MobileNetV3 dalam jaringan saraf konvolusional. Di Indonesia, teknologi kesehatan memainkan peran penting dalam industri kosmetik dan perawatan pribadi, namun, kurangnya pemahaman seringkali menyebabkan konsumen melakukan kesalahan dalam memilih dan menggunakan produk perawatan kulit, yang dapat memperburuk kondisi kulit. Model MobileNetV3, yang telah dilatih sebelumnya, digunakan sebagai dasar model, dan kemudian disempurnakan berdasarkan data gambar perawatan kulit yang spesifik. Hasil penelitian sebelumnya menunjukkan bahwa model ini memiliki akurasi klasifikasi tertinggi (94%) dalam memprediksi penuaan kulit, menjadikannya pilihan unggul dalam penelitian ini. Evaluasi melalui confusion matrix menunjukkan bahwa model menunjukkan performa terbaik pada skenario utama, sementara penurunan performa terjadi pada skenario lainnya. Namun, skenario 4 menunjukkan performa yang mendekati skenario utama, mengindikasikan kemiripan data antara kedua skenario tersebut. Dengan demikian, penelitian ini diharapkan dapat membantu konsumen Indonesia memilih produk perawatan kulit yang tepat dan meningkatkan kesadaran tentang pentingnya penggunaan produk yang sesuai dengan kebutuhan dan jenis kulit mereka.

### **Kata kunci:**

*Convolutional Neural Network, Transfer Learning, MobileNetV3, Klasifikasi, Perawatan Kulit, Kesadaran Konsumen*

## PENDAHULUAN

Teknologi kesehatan memainkan peran penting dalam industri kosmetik dan perawatan pribadi di Indonesia, mendorong pertumbuhan yang signifikan dan membuka peluang bagi pelaku pasar global dan lokal. Konsumen Indonesia menunjukkan peningkatan permintaan terhadap produk kecantikan internasional dan lokal dalam beberapa tahun terakhir, sejalan dengan tren peningkatan pendapatan di pasar tersebut. Namun, kurangnya pemahaman seringkali menyebabkan konsumen melakukan kesalahan dalam memilih dan menggunakan produk perawatan kulit. Kesalahan ini dapat memperburuk kondisi kulit karena penggunaan produk yang tidak sesuai.

Penelitian ini bertujuan untuk mengatasi tantangan tersebut dengan menggunakan pembelajaran transfer dan MobileNetV3 dalam jaringan saraf konvolusional. MobileNetV3 merupakan arsitektur pembelajaran mendalam yang menjanjikan dalam tugas klasifikasi gambar, terutama pada perangkat seluler dan tertanam. Pembelajaran transfer melibatkan penggunaan model MobileNetV3 yang telah dilatih sebelumnya dan menyempurnakannya berdasarkan data gambar perawatan kulit tertentu. Hal ini memungkinkan pengembangan model yang lebih efisien dan akurat untuk membantu konsumen membuat pilihan perawatan kulit berdasarkan kebutuhan dan preferensi pribadi mereka.

Penelitian "Skin Microstructure Segmentation and Aging Classification Using CNN-Based Models" oleh C.-I. Moon dan Onseok Lee pada tahun 2021 menunjukkan bahwa MobileNetV3-Small memiliki akurasi klasifikasi tertinggi (94%) dalam memprediksi penuaan kulit. Model ini menggunakan teknologi terbaru, seperti fungsi aktivasi h-swish, yang mengurangi latensi dan meningkatkan akurasi dibandingkan dengan model sebelumnya. Dibandingkan dengan MobileNetV2, NASNet-Mobile, dan EfficientNet-B0, MobileNetV3-Small lebih cepat dan lebih akurat dalam klasifikasi, menjadikannya pilihan unggul dalam penelitian ini, sehingga diharapkan dapat membantu konsumen Indonesia memilih produk perawatan kulit yang tepat dan meningkatkan kesadaran tentang pentingnya penggunaan produk yang sesuai dengan kebutuhan dan jenis kulit mereka.

## TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Pengembangan teknologi untuk aplikasi visi mobile dan embedded telah menjadi fokus utama dalam beberapa tahun terakhir. Studi terdahulu, seperti "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" oleh Howard et al. (2017), membuktikan potensi besar dari arsitektur MobileNets yang ringan dan efisien dalam hal biaya komputasi dan ukuran model. Arsitektur MobileNets didasarkan pada convolutional terpisah secara depthwise yang memungkinkan pengurangan latensi dan jumlah parameter, menjadikannya ideal untuk aplikasi dengan sumber daya terbatas seperti perangkat mobile. Dengan pengenalan hyperparameter global yang menyeimbangkan latensi dan akurasi, MobileNets memberikan fleksibilitas bagi pengembang model untuk menyesuaikan ukuran model dengan persyaratan aplikasi mereka. Penelitian lanjutan, seperti "Thin MobileNet: An Enhanced MobileNet Architecture" oleh Sinha & El-Sharkawy (2019), telah memperkuat

kinerja MobileNets dengan meningkatkan akurasi dan mengurangi ukuran model. Selain itu, penelitian terbaru, seperti "Skin Microstructure Segmentation and Aging Classification Using CNN-Based Models" oleh Moon & Lee (2021), menegaskan keunggulan MobileNetV3-Small dalam klasifikasi penuaan kulit dibandingkan dengan model-model lainnya seperti NASNet-Mobile dan EfficientNet-B0. Temuan ini menunjukkan bahwa MobileNets terus memimpin dalam pengembangan model yang efisien dan adaptable untuk aplikasi visi mobile dan embedded.

## 2.2 Computer Vision

*Computer vision* adalah disiplin ilmu yang mempelajari bagaimana komputer dapat memahami dan mengolah isi dari sebuah gambar atau video digital (Pratama, 2020). Computer vision sangat berguna untuk mengenali pola dalam gambar untuk keperluan klasifikasi, deteksi objek, segmentasi, dan lain sebagainya.

## 2.3 Convolutional Neural Network

*Convolutional Neural Network* (CNN) adalah jenis *artificial intelligence* yang paling banyak digunakan dalam computer vision (Kusumo, 2020). CNN terdiri dari layer *convolutional* yang berguna untuk mengekstraksi fitur, *layer pooling* yang mengurangi dimensi fitur, dan *fully connected* layer yang mengklasifikasikan fitur. CNN telah banyak digunakan untuk mengenali pola citra medis, deteksi objek, dan lainnya.

## 2.4 Transfer Learning

*Transfer learning* merupakan teknik di mana model yang telah dilatih pada satu tugas ditransfer dan digunakan sebagai starting point untuk tugas lain. Hal ini biasanya dilakukan ketika jumlah data latih untuk tugas baru terbatas (Putra dan Widiyanto, 2018).

*Transfer learning* adalah paradigma potensial yang digunakan untuk memperoleh informasi yang diperoleh dalam konteks sebelumnya ketika jumlah data yang dibutuhkan untuk melatih jaringan pembelajaran mendalam tidak mencukupi dan mengumpulkan data yang tergolong mahal. Model pra-latih menerima data input untuk ekstraksi fitur, dan nilai aktivasi dari beberapa lapisan disimpan dan digunakan sebagai fitur. Lapisan akhir model mengambil ciri-ciri dataset yang baru ditambahkan sementara lapisan awal model tetap dipertahankan untuk *fine-tuning*. Dengan cara ini, tugas baru yang sebanding diimplementasikan ke dalam model pembelajaran mendalam. Berat model disesuaikan sesuai dengan dataset yang baru. Dibandingkan dengan pelatihan jaringan baru, transfer learning lebih cepat karena tidak ada parameter yang dihitung dari awal. Para penulis

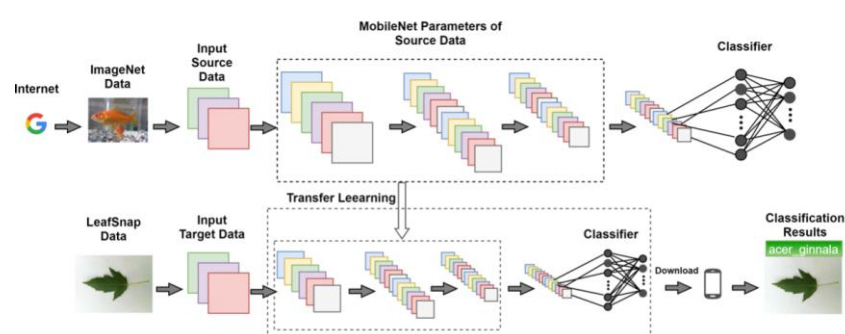
menganalisis berbagai teknik transfer learning dan menyimpulkan bahwa, ketika dibandingkan dengan strategi transfer learning lainnya seperti yang didasarkan pada ruang fitur dan domain yang diberikan, dapat meningkatkan efisiensi dari model deep learning (Hussain et al., 2021).

Gambar 1. Implementasi Transfer Learning MobileNetV3

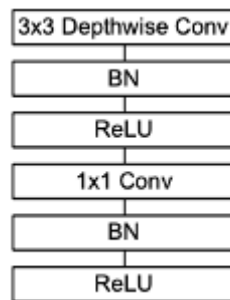
## 2.5 MobileNetV3

Dengan bantuan algoritma NetAdapt dan *network architecture search* (NAS) pada perangkat keras, MobileNetV3 adalah jaringan saraf konvolusi yang dioptimalkan untuk CPU ponsel. MobileNetV3 kemudian diperkuat oleh perkembangan arsitektur baru. Strategi pencarian komplementer, versi nonlinearitas yang baru sehingga efektif untuk lingkungan mobile, dan arsitektur jaringan yang efektif baru sebagai improvisasi versi sebelumnya (Howard et al., 2019).

Arsitektur MobileNet dicirikan oleh *batchnorm* dan nonlinier ReLU. Model ini



memiliki 28 lapisan, dengan 95% waktu komputasi dihabiskan dalam konvolusi  $1 \times 1$  dan 75% parameter berada pada lapisan yang terhubung sepenuhnya. Model ini dilatih di TensorFlow menggunakan RMSprop dan penurunan gradien asinkron yang mirip dengan Inception V3. Namun, model MobileNet menggunakan lebih sedikit teknik regularisasi dan anotasi data, sehingga mengurangi overfitting dan distorsi. Filter *depthwise* memiliki penurunan bobot yang minimal, dan semua model dilatih dengan parameter pelatihan yang sama, berapa pun ukuran modelnya (Howard et al., 2019).



Gambar 2. Konvolusi dengan lapisan Depthwise, batchnorm, dan ReLU.

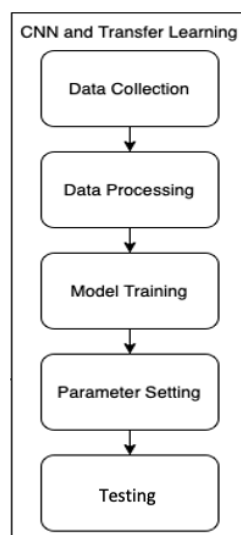
MobileNetV3 merupakan arsitektur convolutional neural network (CNN) yang dirancang khusus untuk perangkat mobile dengan tujuan meningkatkan akurasi klasifikasi gambar pada perangkat dengan daya komputasi terbatas. MobileNetV3 menggunakan blok *bottleneck* residual untuk meningkatkan akurasi tanpa menambah kompleksitas model secara signifikan (Pratama et al., 2019).

Tabel 1. Perbandingan Akurasi Model MobileNet dengan Model Lain.

Model	Accuracy	Recall	Precision	F1-score
NASNet-Mobile	0.92	0.92	0.92	0.92
MobileNetV2	0.93	0.93	0.92	0.92
MobileNetV3-Small	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>
EfficientNet-B0	0.93	0.94	0.93	0.93

Keempat hasil indikator menunjukkan bahwa model MobileNetV3-Small menunjukkan kinerja klasifikasi yang superior dibandingkan dengan model lainnya, seperti NASNet-Mobile, MobileNetV2, dan EfficientNet-B0, sebagaimana yang telah dijelaskan dalam penelitian sebelumnya (Moon C. et al., 2021).

## METODE



Gambar 3. Diagram CNN *Transfer Learning*.

Untuk memulai proses pengembangan suatu sistem atau model, langkah pertama yang harus dilakukan adalah mengumpulkan data yang diperlukan, tahap ini dikenal sebagai *Data Collection*. Setelah data terkumpul, langkah berikutnya adalah memproses data tersebut agar siap digunakan dalam proses pelatihan model, proses ini dikenal sebagai data processing. Setelah data diproses, tahap selanjutnya adalah menentukan parameter dan mengatur model yang akan digunakan dalam pelatihan, langkah ini disebut model training parameter setting. Setelah proses pelatihan selesai dan model telah berhasil dikembangkan, langkah terakhir adalah melakukan deployment, yaitu mengimplementasikan model ke dalam lingkungan produksi sehingga siap digunakan untuk memecahkan masalah yang dihadapi. Dengan demikian, empat tahap tersebut, yaitu *Data Collection*, *data processing*, *model training parameter setting*, dan *deployment*, membentuk rangkaian proses yang penting dalam pengembangan sistem atau model.

### 3.1 Data Collection

Tahap pertama adalah mengumpulkan data yang cukup dan relevan dengan prediksi yang akan dilakukan oleh model. Menentukan jenis data dan jumlah minimum data yang dibutuhkan. Sumber data harus sesuai dengan kasus yang diselesaikan. Setelah data terkumpul, dilakukan labeling atau penandaan data agar sesuai dengan kelas prediksi model. Data kemudian dibagi menjadi dataset latih, validasi, dan uji.

Dataset yang dipakai di penelitian ini yaitu dataset *skin\_type* dan dataset *skin\_disease*. Kedua dataset ini diperoleh secara mandiri dari berbagai platform online. Terdapat empat kelas untuk tipe kulit wajah, masing-masing terdiri dari 110 gambar, dan kelas tersebut adalah Dry, Normal, Oily, dan Sensitive. Sementara itu, empat kelas untuk masalah kulit

wajah terdiri dari 400 gambar, dan kelas tersebut adalah Acne, Black Spots, Puff Eyes, dan Wrinkles.

Gambar 4. Sampel Dataset.

### 3.2 Data Preprocessing

Data yang diakuisisi dibagi atau dibagi menjadi data pelatihan, validasi, dan pengujian, dengan total 440 data jenis kulit dan 1600 data penyakit kulit. Himpunan data pelatihan



digunakan untuk melatih atau membangun model. Himpunan data validasi digunakan untuk mengoptimasi saat melatih model. Model dilatih dengan himpunan pelatihan, kemudian kinerjanya diuji dengan himpunan validasi. Himpunan data pengujian digunakan untuk menguji model setelah proses latihan selesai. Akuisisi data dilakukan melalui program Python, yang memiliki kemampuan untuk membagi secara otomatis untuk mempersingkat waktu. Dalam penelitian ini, hasil pembagian atau pembagian data disimpan pada masing-masing folder yang disebut sebagai data pelatihan, data validasi, dan data tes. Rasio pembagian data adalah 80%, 15%, dan 5%.

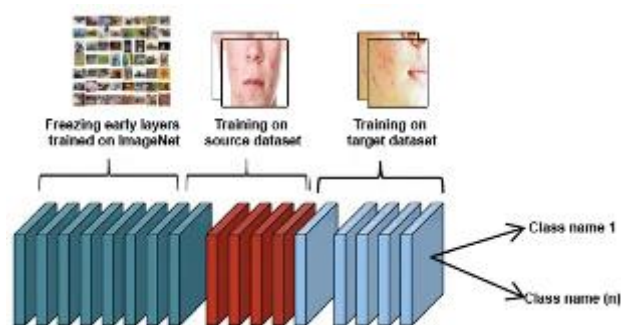
Proses selanjutnya adalah augmentasi gambar menggunakan ImageDataGenerator, yang bertujuan untuk memperluas dataset pelatihan dengan variasi gambar melalui rotasi, pergeseran, pembesaran, dan lainnya. Ini membantu mencegah overfitting dan meningkatkan kemampuan model untuk menggeneralisasi. Argumen dalam ImageDataGenerator seperti 'rescale', 'rotation\_range', 'width\_shift\_range', dan sebagainya, digunakan untuk menyesuaikan gambar. Dengan menggunakan argumen-argumen ini, variasi data yang diberikan ke model selama pelatihan dapat ditingkatkan, yang pada gilirannya membantu meningkatkan kinerja dan generalisasi model.

### 3.3 Hyperparameter Setting

Untuk mendapatkan hasil akurasi yang optimal dan mencegah overfitting pada model, diperlukan penyetelan beberapa parameter dalam arsitektur model. Hyperparameter yang sedang diuji mencakup pengaturan input\_shape, dropout\_rate, dense\_layer, learning\_rate, epoch\_size dan batch\_size yang digunakan. embedding\_dim menentukan bentuk input model, sementara dropout\_rate mencegah overfitting dengan mengontrol proporsi unit yang dinonaktifkan secara acak. Lapisan dense digunakan untuk menggabungkan hasil dari beberapa lapisan dan neuron\_size mempengaruhi kompleksitas korelasi yang dipelajari model. Learning\_rate mengontrol langkah pembelajaran algoritma, sementara epoch\_size dan batch\_size menentukan jumlah iterasi dan ukuran sampel selama pelatihan. Proses penyetelan ini memerlukan eksperimen yang teliti guna menemukan kombinasi parameter terbaik yang dapat meningkatkan performa model secara keseluruhan.

### 3.4 Model Training

Model MobileNetV3 yang sudah dilatih sebelumnya digunakan sebagai basis model dan dilatih ulang dengan data latih yang sudah dipersiapkan. Teknik transfer learning ini memungkinkan pelatihan model yang akurat tanpa memerlukan data dalam jumlah besar. Pada fase ini, proses transfer learning pada MobileNetV3 model akan dilakukan pada gambar sampel kulit sudah dikumpulkan dalam kumpulan data sumber, seperti yang ditunjukkan pada Gambar 4. Langkah ini penting untuk konvergensi domain antara domain sumber dan domain target, dan untuk mengurangi efek dataset ImageNet. Langkah ini penting untuk mengekstraksi fitur yang dekat dengan dataset target. Setelah dilakukan pelatihan terhadap gambar klasifikasi target akan menghasilkan kelas-kelas yang telah ditentukan.



Gambar 5. Proses *Transfer Learning* pada MobileNetV3.

### 3.5 Pengujian

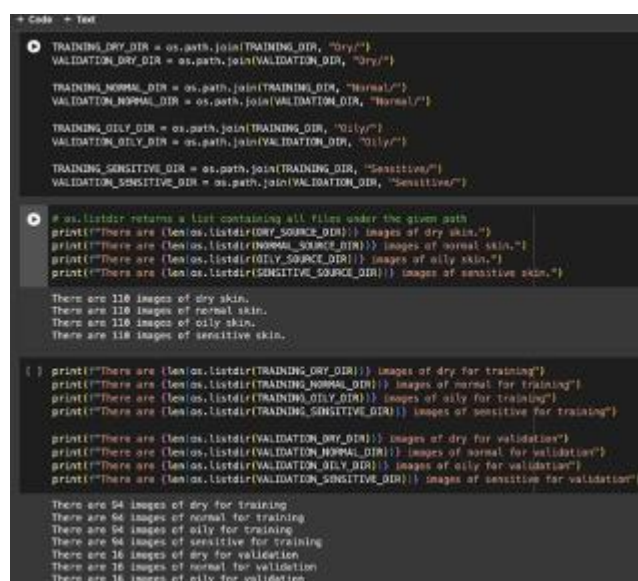
Pengujian dilakukan dengan membuat beberapa skenario yang berbeda, yang memungkinkan pengujian mencakup berbagai kemungkinan yang relevan. Dengan melibatkan sejumlah skenario, pengujian menjadi lebih komprehensif dan memastikan bahwa model atau sistem machine learning diuji secara menyeluruh untuk mendapatkan hasil terbaik. Berbagai skenario yang akan dilakukan mencakup berbagai situasi yang mungkin terjadi.

1. Skenario 1: Mengubah komposisi dataset menjadi 80:10:10.
2. Skenario 2: Tanpa dilakukan *fine-tuning*.
3. Skenario 3: Mengubah optimizer (Adam).
4. Skenario 4: Tanpa *Early Stopping*.
5. *Confusion matrix*

## HASIL DAN PEMBAHASAN

### 4.1 Data Collection

Dataset yang digunakan diperoleh dari berbagai sumber, termasuk open-source Kaggle dan DermNet. Hasil dari pengumpulan dataset, yang terdiri dari sampel gambar dan label-label yang berkaitan. Dataset ini mencakup berbagai masalah kulit seperti *acne*, *black spot*, *puff eyes*, *wrinkle* dengan masing-masing memiliki 400 sampel. Selain itu, dataset juga mencakup berbagai jenis kulit seperti *oily*, *sensitive*, *normal*, dan *dry* dengan masing-masing memiliki 110 sampel. Penggunaan dataset yang luas ini penting dalam melatih model untuk mengidentifikasi dan mengklasifikasikan berbagai masalah kulit serta jenis kulit, yang dapat membantu dalam pengembangan solusi di bidang dermatologi dan perawatan kulit secara umum.



```
Code + Text
TRAINING_DRY_DIR = os.path.join(TRAINING_DIR, "dry/")
VALIDATION_DRY_DIR = os.path.join(VALIDATION_DIR, "dry/")

TRAINING_NORMAL_DIR = os.path.join(TRAINING_DIR, "normal/")
VALIDATION_NORMAL_DIR = os.path.join(VALIDATION_DIR, "normal/")

TRAINING_OILY_DIR = os.path.join(TRAINING_DIR, "oily/")
VALIDATION_OILY_DIR = os.path.join(VALIDATION_DIR, "oily/")

TRAINING_SENSITIVE_DIR = os.path.join(TRAINING_DIR, "sensitive/")
VALIDATION_SENSITIVE_DIR = os.path.join(VALIDATION_DIR, "sensitive/")

# os.listdir returns a list containing all files under the given path
print("There are", len(os.listdir(DRY_SOURCE_DIR)), "images of dry skin.")
print("There are", len(os.listdir(NORMAL_SOURCE_DIR)), "images of normal skin.")
print("There are", len(os.listdir(OILY_SOURCE_DIR)), "images of oily skin.")
print("There are", len(os.listdir(SENSITIVE_SOURCE_DIR)), "images of sensitive skin.")

There are 110 images of dry skin.
There are 110 images of normal skin.
There are 110 images of oily skin.
There are 110 images of sensitive skin.

() print("There are", len(os.listdir(TRAINING_DRY_DIR)), "images of dry for training")
print("There are", len(os.listdir(TRAINING_NORMAL_DIR)), "images of normal for training")
print("There are", len(os.listdir(TRAINING_OILY_DIR)), "images of oily for training")
print("There are", len(os.listdir(TRAINING_SENSITIVE_DIR)), "images of sensitive for training")

print("There are", len(os.listdir(VALIDATION_DRY_DIR)), "images of dry for validation")
print("There are", len(os.listdir(VALIDATION_NORMAL_DIR)), "images of normal for validation")
print("There are", len(os.listdir(VALIDATION_OILY_DIR)), "images of oily for validation")
print("There are", len(os.listdir(VALIDATION_SENSITIVE_DIR)), "images of sensitive for validation")

There are 54 images of dry for training
There are 54 images of sensitive for training
There are 54 images of oily for training
There are 54 images of normal for training
There are 16 images of dry for validation
There are 16 images of normal for validation
There are 16 images of oily for validation
```

Gambar 6. Proses *Data Collection*.

Selain itu, distribusi data latihan, validasi, dan uji juga diatur sesuai dengan standar praktik. Data latihan disediakan sebanyak 80% dari total dataset, sementara 15% disisihkan untuk data validasi, dan 5% digunakan sebagai data uji. Pembagian ini penting untuk memastikan bahwa model yang dikembangkan memiliki performa yang baik dalam berbagai situasi, termasuk kemampuannya untuk menggeneralisasi pola dari data yang belum pernah dilihat sebelumnya, sehingga penerapan pembagian data pada Gambar 6 dilakukan. Dengan mengalokasikan data dengan proporsi yang tepat untuk setiap tahap pengembangan model, proses evaluasi dan validasi dapat dilakukan secara efektif, sehingga hasil yang akurat dan andal dapat dihasilkan.

#### 4.2 Data Preprocessing

Proses Augmentasi Data merupakan tahap penting dalam pra-pemrosesan dataset yang dilakukan sebelum pemodelan. Augmentasi data dilakukan untuk meningkatkan variasi data latihan dengan cara menghasilkan versi yang sedikit berbeda dari gambar asli. Hal ini bertujuan untuk memperluas dataset yang tersedia, sehingga model dapat lebih baik dalam mengenali

pola yang mungkin muncul dalam berbagai situasi. Dalam konteks ini, penggunaan ImageDataGenerator memungkinkan berbagai transformasi gambar, seperti rotasi, pergeseran, pembesaran, pencerminan horizontal, dan lainnya. Rescale digunakan untuk menormalisasi nilai piksel dalam gambar, sementara argumen lainnya, seperti rotation\_range, width\_shift\_range, height\_shift\_range, shear\_range, zoom\_range, horizontal\_flip, dan fill\_mode, memberikan kontrol terhadap variasi yang dihasilkan. Dengan augmentasi data yang tepat, model dapat lebih robust dan dapat memberikan hasil yang lebih baik saat diterapkan pada data yang belum pernah dilihat sebelumnya. Berikut adalah contoh kode penerapan augmentasi data menggunakan Keras ImageDataGenerator.

Gambar 7. Kode Augmentasi.

Selanjutnya membuat model dengan menggunakan arsitektur yang sudah dilatih sebelumnya (*pre-trained model*). Pada contoh ini, kita menggunakan model

```
def train_val_generators(TRAINING_DIR, VALIDATION_DIR):  
    # Instantiate the ImageDataGenerator class  
    train_datagen = ImageDataGenerator(rescale = 1./255,  
                                      rotation_range=40,  
                                      width_shift_range=0.2,  
                                      height_shift_range=0.2,  
                                      shear_range=0.1,  
                                      zoom_range=0.1,  
                                      horizontal_flip=True,  
                                      fill_mode='nearest')  
    val_datagen = ImageDataGenerator(rescale = 1./255,  
                                    rotation_range=40,  
                                    width_shift_range=0.2,  
                                    height_shift_range=0.2,  
                                    shear_range=0.1,  
                                    zoom_range=0.1,  
                                    horizontal_flip=True,  
                                    fill_mode='nearest')  
  
    train_generator = train_datagen.flow_from_directory(TRAINING_DIR,  
                                                      target_size=(224, 224),  
                                                      batch_size=32,  
                                                      class_mode='categorical')  
    val_generator = val_datagen.flow_from_directory(VALIDATION_DIR,  
                                                  target_size=(224, 224),  
                                                  batch_size=32,  
                                                  class_mode='categorical')  
  
    return train_generator, val_generator  
  
def create_pre_trained_model():  
    pre_trained_model = tf.keras.applications.MobileNetV3Large(input_shape=(224, 224, 3),  
                                                             include_top=False,  
                                                             weights='imagenet')  
  
    # Make all the layers in the pre-trained model non-trainable  
    from tensorflow.keras import layers  
    for layer in pre_trained_model.layers:  
        layer.trainable = False  
  
    return pre_trained_model  
  
pre_trained_model = create_pre_trained_model()  
  
# Print the model summary  
pre_trained_model.summary()
```

MobileNetV3Large yang telah dilatih dengan dataset ImageNet. Model ini telah terbukti efektif dalam mempelajari fitur-fitur umum dari gambar. Berikut adalah kode untuk membuat model *pretrained*.

Gambar 8. Kode *Pretrained Model*.

Kode tersebut menghasilkan luaran berupa informasi tentang jumlah total parameter dalam model dan jumlah parameter yang dapat dilatih ulang (trainable parameters). Dalam hal ini, terdapat total 2.996.352 parameter dalam model tersebut. Namun, tidak ada parameter yang dapat dilatih ulang, sehingga jumlah trainable parameters adalah 0. Hal ini menunjukkan bahwa model yang digunakan merupakan model yang sudah di-pretrained dan parameter-parameter tersebut tidak akan diubah selama proses pelatihan.

### 4.3 Hyperparameter Setting

Selanjutnya model dibangun menggunakan lapisan-lapisan berurutan (Sequential Layer). Model ini memanfaatkan arsitektur MobileNetV3 sebagai model dasar yang telah

dipelajari sebelumnya. Arsitektur MobileNetV3 digunakan untuk mengekstraksi fitur-fitur dari gambar. Berikut adalah contoh penerapan lapisan berurutan.

Gambar 9. Kode *fine-tuning*.

```
# build model
model = tf.keras.Sequential([
    pre_trained_model,
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(1e-5)),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(4, activation='softmax')
])

model.build([None, 224, 224, 3])

model.compile([
    optimizer=tf.keras.optimizers.SGD(learning_rate=1e-5, momentum=0.9),
    loss='categorical_crossentropy',
    metrics=['accuracy']
])
```

#### 4.4 Model Training

Tahap pelatihan adalah tahap terakhir sebelum model dapat digunakan untuk tujuan yang dimaksud. Pada tahap ini, model yang telah menjalani proses *fine-tuning* akan dilatih menggunakan dataset yang telah disiapkan sebelumnya. Berikut ini adalah kode pelatihan model.

```
[ ] # Define a Callback class that stops training once accuracy reaches 85.0%
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(
            logs.get('accuracy') is not None and
            logs.get('accuracy') > 0.85 and
            logs.get('val_accuracy') > 0.85 and
            logs.get('accuracy') - logs.get('val_accuracy') < 0.05
        ):
            print("\nReached 85% accuracy and validation accuracy so cancelling training!")
            self.model.stop_training = True

[ ] callbacks = myCallback()
history = model.fit(train_generator,
                    validation_data = validation_generator,
                    epochs=100,
                    verbose=1,
                    callbacks=[callbacks]
                    )
```

Gambar 10. Kode *Model Training*.

#### 4.5 Pengujian

Tabel 2. Hasil Pengujian *Confusion Matrix*.

Skenario	Model	<i>F1 Score</i>	<i>Recall</i>	<i>Precision</i>	<i>Accuracy</i>
Skenario Utama	SkinTypes	0.853	0.854	0.854	0.854
	SkinDiseases	0.843	0.842	0.851	0.841
Skenario 1	SkinTypes	0.771	0.770	0.778	0.770

	SkinDiseases	0.772	0.774	0.783	0.775
Skenario 2	SkinTypes	0.704	0.708	0.711	0.708
	SkinDiseases	0.741	0.745	0.752	0.744
Skenario 3	SkinTypes	0.747	0.750	0.750	0.750
	SkinDiseases	0.729	0.734	0.744	0.732
Skenario 4	SkinTypes	0.833	0.833	0.834	0.833
	SkinDiseases	0.831	0.829	0.839	0.828

Performa model pada skenario utama menunjukkan hasil yang sangat baik, dengan nilai F1 score, recall, precision, dan accuracy yang tinggi, semua di atas 0.84, baik untuk klasifikasi *SkinTypes* maupun *SkinDiseases*. Namun, pada skenario lain, terlihat penurunan performa pada kedua model. Penurunan ini paling signifikan terjadi pada Skenario 2, sementara penurunan paling kecil terjadi pada Skenario 4.

Dari perbandingan skenario, Skenario 1 menunjukkan performa model yang sedikit lebih rendah daripada skenario utama, sementara Skenario 2 mengalami penurunan performa yang signifikan, menandakan bahwa data pada skenario tersebut mungkin lebih sulit untuk diklasifikasikan. Skenario 3 menunjukkan peningkatan performa dibandingkan dengan Skenario 2, meskipun masih lebih rendah dari skenario utama. Di sisi lain, Skenario 4 menunjukkan performa yang mendekati skenario utama, mengindikasikan bahwa data pada skenario tersebut mungkin lebih mirip dengan data pada skenario utama.

Secara keseluruhan, model menunjukkan performa terbaik pada skenario utama, sementara penurunan performa pada skenario lain menunjukkan bahwa data pada skenario-skenario tersebut mungkin lebih sulit untuk diklasifikasikan. Dari analisis ini, dapat disimpulkan bahwa data pada skenario 4 mungkin lebih mirip dengan data pada skenario utama daripada skenario lainnya.

## **PENUTUP**

### **Kesimpulan**

Dari hasil penelitian ini, dapat disimpulkan bahwa teknologi kesehatan memainkan peran penting dalam industri kosmetik dan perawatan pribadi di Indonesia, mendorong pertumbuhan yang signifikan dan membuka peluang bagi pelaku pasar global dan lokal. Meskipun demikian, kurangnya pemahaman konsumen seringkali menyebabkan kesalahan

dalam memilih dan menggunakan produk perawatan kulit. Untuk mengatasi tantangan ini, penelitian ini menggunakan pembelajaran transfer dan MobileNetV3 dalam jaringan saraf konvolusional untuk mengembangkan model yang lebih efisien dan akurat dalam klasifikasi perawatan kulit. Hasil penelitian menunjukkan bahwa model MobileNetV3 memiliki akurasi klasifikasi tertinggi (94%) dalam memprediksi penuaan kulit, menjadikannya pilihan unggul dalam penelitian ini. Evaluasi melalui *confusion matrix* menunjukkan bahwa model menunjukkan performa terbaik pada skenario utama, sementara penurunan performa terjadi pada skenario lainnya. Namun, skenario 4 menunjukkan performa yang mendekati skenario utama, mengindikasikan kemiripan data antara kedua skenario tersebut. Oleh karena itu, penelitian ini diharapkan dapat membantu konsumen Indonesia memilih produk perawatan kulit yang tepat dan meningkatkan kesadaran tentang pentingnya penggunaan produk yang sesuai dengan kebutuhan dan jenis kulit mereka.

#### DAFTAR PUSTAKA

- C.-I. Moon and O. Lee, "Skin Microstructure Segmentation and Aging Classification Using CNN-Based Models" in IEEE Access, vol. 10, pp. 4327-4339, 2022, doi: 10.1109/ACCESS.2021.3140031.
- Danendra, F. A., Anggraeny, F. T., & Maulana, H. (2023). *Klasifikasi Citra Penyakit Daun Cabai Rawit Dengan Menggunakan CNN Arsitektur AlexNet dan SqueezeNet*. Universitas Singaperbangsa Karawang, Vol 12 No 01 (2023): Mei 2023. Retrieved from <https://journal.unsika.ac.id/index.php/syntax/article/view/7947/3936>
- Halim, S., Rahman, T. K. A., Saragih, H., & Rahmat, B. (2022). *Poor Population Classification System Using Convolutional Neural Network (CNN)*. Proceeding - IEEE 8th Information Technology International Seminar, ITIS 2022, 11–14. <https://doi.org/10.1109/ITIS57155.2022.10010285>
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). *Searching for MobileNetV3*. Google AI, Google Brain.
- Hussain, A., Barua, B., Osman, A., Abozariba, R., & Asyhari, A. T. (2021). *Performance of MobileNetV3 Transfer Learning on Handheld Device-based Real-Time Tree Species Identification*. In 2021 26th International Conference on Automation and Computing (ICAC) (pp. 1-6). IEEE. DOI: 10.23919/ICAC50006.2021.9594222
- Kusumo, A. N. (2020). *Penerapan Convolutional Neural Network untuk Prediksi Penyakit Kulit*. Jurnal Sains dan Informatika, 6(2), 106-115.
- Putra, R.E., Widiyanto, A.D., 2018. *Implementasi Transfer Learning Dengan Arsitektur MobileNet Pada Klasifikasi Jamur*. Jurnal Teknik Informatika dan Sistem Informasi, 4(3), pp.453-462.
- Pratama, I.G.Y.E., Sugiarta, I.W., & Yuniarta, I.D.P.S. (2019). *Sistem Klasifikasi Gulma pada Tanaman Padi Menggunakan Deep Convolutional Neural Network*. Jurnal Ilmu Komputer dan Agroteknologi, 5(2), 29-36.

- Pratama, A. R. (2020). *Penerapan Computer Vision untuk Deteksi Objek Menggunakan OpenCV Python*. Jurnal Ilmiah Teknologi Informasi Asia, 14(2), 91-98.
- Pratama, I.G.Y.S., Purwarianti, A., Pratama, M.R.H.A., 2019. *Klasifikasi Gambar Tanaman Padi Menggunakan Convolutional Neural Network Dengan Metode Transfer Learning*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 3(2), pp.1689-1698.
- Sulthoni, S., & Kusumawardani, D. (2018). *Penerapan Deep Learning untuk Prediksi Harga Saham Menggunakan LSTM*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(10), 3148-3156.
- Sinha, D., & El-Sharkawy, M. (2019). *Thin MobileNet: An Enhanced MobileNet Architecture*. In 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 1-6). IEEE. DOI: 10.1109/UEMCON47517.2019.8993089.