

Acne Severity Classification Study Using Convolutional Neural Network Algorithm with MobileNetV2 Architecture

Kajian Klasifikasi Tingkat Keparahan Jerawat Menggunakan Algoritma Convolutional Neural Network

Faadiyah Ramadhani^{1‡}, Septian Rahardiantoro¹, Mohammad Masjkur¹

¹Department of Statistics, IPB University, Indonesia

[‡]corresponding author: faadiyahrrfaadiyah@apps.ipb.ac.id

Copyright © 2024 Faadiyah Ramadhani, Septian Rahardiantoro, and Mohammad Masjkur. This is an open-access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Data classification is a key technique in machine learning that maps patterns and features of input data into a target class. Significant developments in data classification occur in deep learning with neural networks and Convolutional Neural Networks (CNN) that are able to extract image features automatically. CNN can classify the level of a condition based on image data, one of which is the severity of acne. Acne (acne vulgaris) is a common skin disease with varying severity. This study aims to apply the CNN MobileNetV2 model to classify acne severity based on acne input images. The data consists of 1457 acne images at 4 severity levels divided into 80% training data and 20% test data. MobileNetV2 was used as a feature extractor through transfer learning. Fine-tuning and classification were performed using fully connected layers with ReLU and softmax activation functions. The model was evaluated with a confusion matrix and classification report. The model with a combination of hyperparameter batch size 16 and a learning rate of 0.00001 was the best model that achieved 87.29% accuracy with 89% precision, 84% recall, and 86% F1 score for classifying acne severity.

Keywords: acne vulgaris, classification, CNN, mobilenetv2, severity.

1. Pendahuluan

Klasifikasi data merupakan salah satu teknik utama dalam machine learning yang berfokus pada pengategorian atau pengelompokan data ke dalam kelas-kelas yang telah ditentukan (Han *et al.*, 2011). Teknik ini termasuk ke dalam jenis *supervised learning*, di mana terdapat data latih yang sudah diberi label kelas untuk melatih model klasifikasi (Jordan dan Mitchell, 2015). Berdasarkan data latih tersebut, model klasifikasi belajar untuk memetakan pola, atribut, serta fitur-fitur dari data input ke

dalam kelas spesifik yang sudah didefinisikan, seperti *spam* atau bukan *spam*, normal atau abnormal, sehat atau sakit, dan lain sebagainya.

Terdapat beberapa metode yang dapat digunakan untuk melakukan klasifikasi, antara lain Regresi Logistik, Decision Tree, k-Nearest Neighbor (KNN), Naïve Bayes, Support Vector Machine (SVM) hingga metode *deep learning*. Perkembangan signifikan klasifikasi data terjadi pada *deep learning* yang memanfaatkan *neural network* dengan banyak *hidden layer* untuk klasifikasi data gambar (Lecun *et al.*, 2015). Salah satu metode populer untuk mengklasifikasikan gambar adalah Convolutional Neural Network (CNN) yang memiliki kemampuan melakukan ekstraksi fitur secara otomatis dari data gambar melalui proses konvolusi dan *pooling* tanpa memerlukan fitur *engineering manual*. CNN mampu menangkap pola-pola kompleks pada berbagai tingkat abstraksi gambar. Menurut Sandler *et al.*, (2018) salah satu arsitektur CNN yang efisien dan populer adalah MobileNetV2, yang dirancang untuk perangkat dengan daya komputasi terbatas namun tetap mempertahankan akurasi yang tinggi. MobileNetV2 dapat digunakan sebagai *pretrained model*, di mana pengetahuan yang diperoleh dari pelatihan pada set data besar seperti ImageNet dapat ditransfer untuk tugas klasifikasi baru melalui teknik *transfer learning*.

Jerawat (*acne vulgaris*) merupakan penyakit kulit umum akibat produksi sebum berlebih pada kelenjar minyak dan penyumbatan pori-pori oleh sel kulit mati (Hasanah dan Hasan, 2022). Jerawat memiliki prevalensi yang tinggi, mencapai sekitar 85% dari populasi pada usia 11 sampai 30 tahun di seluruh dunia (Okoro *et al.*, 2016). Jerawat juga menjadi permasalahan kulit yang umum dan terus meningkat di Indonesia, terutama pada kalangan remaja. Studi menunjukkan 69.7% remaja perempuan dan 30.3% remaja laki-laki menderita jerawat, dengan tingkat kejadian tertinggi, yaitu 53.2% terjadi pada usia 16–25 tahun (Sibero *et al.*, 2019). Jerawat memiliki tingkat keparahan yang bervariasi, mulai dari jerawat yang ringan hingga berat (Melisa *et al.*, 2022). Selain prevalensi yang tinggi, tantangan lainnya muncul akibat keterbatasan akses terhadap perawatan profesional untuk jerawat karena biaya yang tinggi dan tingkat pengetahuan masyarakat yang rendah. Seringkali, masyarakat mencari informasi di internet tanpa ada jaminan kebenaran. Sementara itu, konsultasi langsung dengan ahli perawatan kulit cenderung tidak terjangkau, baik dari segi biaya maupun waktu yang dibutuhkan (Hadianti *et al.*, 2015). Oleh karena itu, diperlukan solusi yang lebih terjangkau dan efisien untuk menangani permasalahan jerawat mereka dengan mengidentifikasi tingkat keparahan jerawat. Dengan metode ekstraksi fitur dan penerapan proses klasifikasi, hasil perawatan dapat dievaluasi secara menyeluruh dan mencapai efektivitas yang optimal (Becker *et al.*, 2017).

Penelitian sebelumnya mengenai klasifikasi jerawat telah dilakukan Achmad *et al.*, (2021) menggunakan empat fitur metode Gray Level Co-occurrence Matrix (GLCM), yakni *contrast*, *correlations*, *energy*, dan *homogeneity*, serta Backpropagation. Penelitian tersebut memperoleh akurasi sebesar 56,67% untuk mengidentifikasi jenis jerawat berdasarkan tekstur. Penelitian lebih lanjut dilakukan oleh Ramadhani *et al.*, (2018) dengan mengintegrasikan lima fitur GLCM, seperti *contrast*, *correlation*, *energy*, *dissimilarity*, dan *entropy* yang meningkatkan akurasi hingga mencapai 72%. Selanjutnya, Rianto dan Listianto (2023) mengambil pendekatan yang berbeda dengan set data yang bersumber dari Kaggle untuk mengklasifikasikan tingkat keparahan jerawat menggunakan CNN dengan label level 0, 1, dan 2 pada data latih, yang masing-masing berisi 500 citra dan menghasilkan akurasi sebesar 63,63% dengan 50 *epochs*.

Dalam konteks klasifikasi tingkat keparahan jerawat, penggunaan MobileNetV2 dengan *transfer learning* memungkinkan adaptasi model yang telah dilatih

sebelumnya untuk menangkap fitur-fitur spesifik dari citra jerawat. Hal ini berpotensi meningkatkan akurasi klasifikasi bahkan dengan set data yang terbatas. Dengan demikian, CNN MobileNetV2 dapat digunakan untuk mengklasifikasikan tingkat keparahan jerawat.

Adapun, tujuan dilakukannya penelitian ini adalah (1) Mengimplementasikan *transfer learning* model CNN MobileNetV2 untuk klasifikasi tingkat keparahan jerawat, dan (2) Mengkaji kombinasi *hyperparameter* terbaik untuk meningkatkan performa model CNN MobileNetV2 dalam klasifikasi tingkat keparahan jerawat.

2. Metodologi

2.1 Bahan dan Data

Penelitian ini menggunakan set data ACNE04 yang dikumpulkan oleh Wu *et al.*, (2019), terdiri dari 1457 citra jerawat yang dikelompokkan dalam empat tingkat keparahan, yaitu level 0 sebanyak 497 citra, level 1 sebanyak 637 citra, level 2 sebanyak 186 citra, dan level 3 sebanyak 137 citra. Level 0 menunjukkan jerawat ringan dengan sedikit lesi, yang merujuk pada perubahan atau kelainan pada jaringan tubuh, seperti komedo. Level 1 menunjukkan jerawat sedang dengan lebih banyak lesi, termasuk beberapa peradangan seperti papul dan pustul yang tampak merah dan bengkak. Level 2 menunjukkan jerawat parah dengan banyak nodul dan peradangan signifikan, sedangkan level 3 menunjukkan jerawat sangat parah dengan banyak lesi meradang, termasuk nodul dan kista. Awalnya, resolusi gambar bervariasi tetapi kemudian disesuaikan menjadi 224×224 *pixel* untuk pelatihan model. Contoh set data ditampilkan dalam Gambar 1.



Level 0

Level 1

Level 2

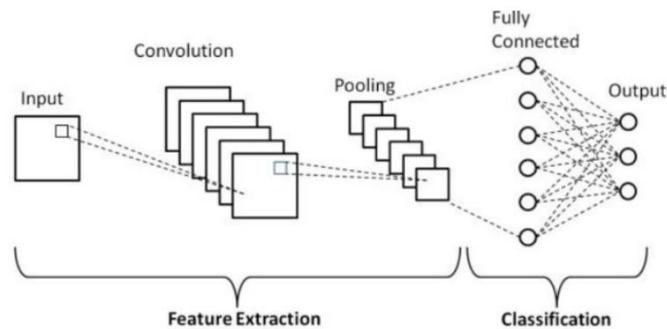
Level 3

Gambar 1: Contoh set data beserta labelnya

2.2 Convolutional Neural Network (CNN)

Deep learning adalah salah satu cabang pembelajaran mesin yang memanfaatkan jaringan saraf tiruan dengan banyak lapisan tersembunyi untuk mempelajari representasi data tingkat tinggi secara bertahap. CNN merupakan algoritma *deep learning* yang paling terkenal dan banyak digunakan dalam bidang *image processing*. CNN mengidentifikasi fitur secara otomatis tanpa supervisi manusia. Struktur CNN terinspirasi dari neuron di otak manusia dan hewan. CNN memanfaatkan representasi yang setara, interaksi yang jarang, dan berbagi parameter untuk memproses data input 2D, seperti citra. Hal ini memungkinkan CNN dilatih dengan jumlah parameter yang jauh lebih sedikit dan kecepatan komputasi yang lebih tinggi dibandingkan jaringan saraf tersambung-penuh (*fully connected*) konvensional. Kelebihan utama CNN ada pada kemampuannya dalam mengekstraksi korelasi lokal pada input citra layaknya sel-sel visual pada korteks visual. CNN terdiri dari sejumlah *layer* dan masing-masing

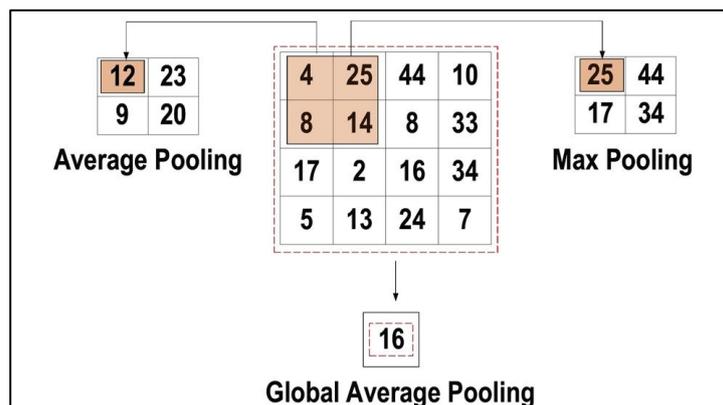
memiliki fungsi filterisasi tertentu dalam setiap prosesnya. Selama proses pelatihan, CNN melalui tiga tahap utama, yakni *convolutional layer*, *pooling layer*, dan *fully connected layer* (Alzubaidi *et al.*, 2021). Algoritma CNN ditampilkan pada Gambar 2.



Gambar 2: Algoritma CNN (Rianto dan Listianto, 2023)

Algoritma CNN terdiri dari beberapa lapisan yang memiliki fungsi spesifik untuk ekstraksi fitur dan klasifikasi. Pada tahap awal, terdapat *convolution layer* yang berperan sangat penting. Lapisan ini terdiri dari sejumlah filter konvolusi atau kernel yang digunakan untuk mengonvolusi citra input. Kernel ini pada awalnya memiliki bobot yang diinisialisasi secara acak, dan selama proses pelatihan, bobot ini diperbarui untuk mengekstraksi fitur-fitur signifikan dari citra input. Proses konvolusi melibatkan kernel yang bergerak melintasi citra input secara horizontal dan vertikal, melakukan perkalian titik antara nilai-nilai citra dan kernel untuk menghasilkan peta fitur *output*.

Setelah *convolution layer*, terdapat *pooling layer* yang bertujuan untuk mengurangi ukuran peta fitur dengan tetap mempertahankan informasi penting. Metode *pooling* yang umum digunakan meliputi *Average Pooling*, *Max Pooling*, dan *Global Average Pooling*. Misalnya, pada *Average Pooling*, nilai rata-rata dari setiap area kernel diambil untuk membentuk peta fitur yang lebih kecil. *Pooling* membantu mengurangi jumlah parameter dan komputasi dalam jaringan, sekaligus mengontrol *overfitting* dengan mengabstraksi representasi fitur. Contoh ilustrasi *pooling layer* ditampilkan pada Gambar 3.



Gambar 3: Ilustrasi metode *pooling* (Alzubaidi *et al.*, 2021)

Lapisan *non-linearitas* atau fungsi aktivasi kemudian diterapkan setelah setiap *convolution* dan *pooling layer*. Fungsi aktivasi seperti ReLU (*Rectified Linear Unit*) mengubah semua nilai input menjadi positif, membantu jaringan menangkap hubungan *non-linear* yang kompleks. Fungsi aktivasi ReLU ditampilkan pada Persamaan 1. Alternatif dari ReLU seperti Leaky ReLU dan Parametric ReLU digunakan untuk mengatasi masalah "Dying ReLU", di mana neuron berhenti berfungsi

jika nilainya negatif.

$$f(x)_{ReLU} = \max(0, x) \quad (1)$$

Kemudian, lapisan *fully connected* digunakan untuk klasifikasi akhir. Dalam lapisan ini, setiap neuron terhubung dengan semua neuron di lapisan sebelumnya, mirip dengan jaringan syaraf tiruan konvensional. Input dari lapisan terakhir *pooling* atau *convolution* diubah menjadi vektor sebelum dimasukkan ke lapisan *fully connected*. Lapisan ini menggabungkan semua fitur yang diekstraksi untuk menghasilkan *output* akhir, seperti klasifikasi citra. Kombinasi dari lapisan-lapisan ini memungkinkan CNN untuk melakukan ekstraksi fitur yang efisien dan akurat serta klasifikasi pada data visual (Alzubaidi et al., 2021).

2.2.1. Hyperparameter CNN

Hyperparameter adalah elemen dalam *deep learning* yang dapat diatur selama proses pelatihan model. Memilih *hyperparameter* yang baik adalah langkah penting yang memengaruhi kinerja model secara keseluruhan. Proses ini dikenal sebagai *hyperparameter tuning*, yaitu mencari *hyperparameter* terbaik dari sekumpulan nilai yang telah ditentukan untuk setiap *hyperparameter*. Beberapa *hyperparameter* yang digunakan dalam pemodelan CNN meliputi:

1. *Epochs* merupakan jumlah siklus pelatihan yang dilakukan pada seluruh set data. Semakin banyak *epochs*, model akan semakin belajar dari data yang diberikan, tetapi dapat menyebabkan *overfitting* jika terlalu banyak.
2. *Batch size* merupakan jumlah data yang diproses secara bersamaan dalam satu iterasi pelatihan. Semakin besar *batch size*, waktu pelatihan menjadi lebih cepat, tetapi dapat menyebabkan bias dan akurasi lebih rendah. Selain itu, pemilihan *batch size* juga berkaitan dengan *learning rate* yang digunakan dalam proses pelatihan.
3. *Learning rate* mengatur besarnya perubahan yang diterapkan selama proses pelatihan. Pemilihan *learning rate* yang tepat sangat penting dalam melatih jaringan syaraf (Sellat et al., 2022). *Learning rate* yang terlalu rendah bisa memperlambat proses pelatihan. Sedangkan nilai yang terlalu tinggi bisa membuat model sulit mencapai konvergensi. Hal ini menunjukkan bahwa *learning rate*, sebagai salah satu *hyperparameter* yang memiliki dampak signifikan terhadap kinerja model CNN (Rochmawanti et al., 2021).
4. *Optimizer* berfungsi untuk menyesuaikan *learning rate* agar model dapat mencapai konvergensi. Salah satu *optimizer* yang sering digunakan dalam CNN adalah *Adaptive Moment Estimation (Adam)*. *Adam*, yang diperkenalkan oleh Kingma dan Ba (2014), adalah sebuah metode optimisasi yang menggunakan *stochastic gradient*. Pendekatannya melibatkan penggunaan rata-rata bergerak (*stochastic gradient*) yang mempertimbangkan momen pertama dan kedua.

2.3 Transfer Learning

Transfer learning pada CNN merujuk pada teknik menggunakan model CNN yang telah dilatih sebelumnya pada kumpulan data yang besar (domain sumber) dan menyetelnya (*fine-tuning*) dengan data baru dari domain target untuk meningkatkan akurasi dan efisiensi pembelajaran. Pendekatan ini sangat bermanfaat ketika data berlabel untuk domain target terbatas, seperti dalam kasus analisis citra medis untuk tugas seperti deteksi jerawat. Dengan mentransfer pengetahuan yang dipelajari dari

domain sumber yang memiliki data berlimpah, *transfer learning* membantu meningkatkan kinerja model pada domain target yang kekurangan data melalui teknik seperti ekstraksi fitur dan adaptasi domain (Albertino, 2023). *Transfer learning* memungkinkan penggunaan kembali (*reuse*) fitur-fitur yang telah dipelajari dari satu domain ke domain lainnya. Ini meningkatkan kemampuan model pada domain target sambil mengurangi kebutuhan untuk mengumpulkan data baru yang banyak dan upaya anotasi data yang ekstensif. Pada CNN, *transfer learning* terbukti efektif dalam berbagai tugas klasifikasi gambar, menawarkan peningkatan akurasi sekaligus efisiensi komputasi dibandingkan melatih model dari awal dengan jumlah data yang terbatas.

2.4 MobileNetV2

MobileNetV2 merupakan arsitektur CNN yang cocok untuk mengatasi batasan daya komputasi saat melakukan tugas, seperti klasifikasi gambar menggunakan *deep learning*. MobileNetV2 menggunakan teknik *depthwise separable convolution*, *linear bottlenecks*, dan *shortcut connections* untuk membangun jaringan yang sangat efisien dengan jumlah parameter yang jauh lebih sedikit dibandingkan arsitektur CNN konvensional, namun tetap mempertahankan akurasi yang tinggi. Hal ini memungkinkan MobileNetV2 untuk melakukan inferensi gambar dan pembelajaran representasi yang cepat dan akurat pada perangkat dengan daya komputasi dan memori terbatas. Selain itu, MobileNetV2 telah dilatih dengan set data ImageNet sebelumnya, sehingga dapat langsung digunakan sebagai ekstraktor fitur citra tanpa perlu melatih ulang seluruh model (Howard *et al.*, 2017).

2.5 Tahapan Penelitian

Analisis pada penelitian ini dilakukan menggunakan *software* Python dengan bantuan *package* Tensorflow sebagai *platform* utama. Tahapan analisis yang dilakukan pada penelitian dimulai dari persiapan data, pembagian data, praproses data, augmentasi, klasifikasi, dan evaluasi.

a. Persiapan Data

Mengunduh dan mengekstrak set data yang akan digunakan. Pada penelitian ini menggunakan data ACNE04 yang diunduh dari laman GitHub dalam format *tarball*. Setelah di unduh, *file* tersebut diekstrak sehingga menjadi sebuah folder yang berisi data citra jerawat dalam format JPG.

b. Pembagian Data

Set data gambar dibagi menjadi 80% data latih sebanyak 1166 gambar dan 20% data uji sebanyak 291 gambar. Dari jumlah tersebut mencakup jumlah data latih dan data uji dari setiap level.

c. Praproses Data

Praproses data merupakan langkah awal pengolahan data untuk mengoptimalkan data sebelum dilakukan pelatihan model. Pada proses ini dilakukan penentuan beberapa *hyperparameter*, seperti *batch size* 8, 16, dan 32, serta *learning rate* 0,1, 0,01, 0,001, 0,0001, dan 0,00001. Jumlah maksimal *epochs* yang digunakan adalah 200 dan dengan menggunakan optimasi Adam. Berdasarkan *hyperparameter* yang digunakan (*batch size* dan *learning rate*)

akan dikombinasikan menjadi 15 model. Kombinasi dari model tersebut dapat dilihat pada Tabel 1. *Hyperparameter* tersebut disesuaikan dengan arsitektur MobileNetV2 yang digunakan. Kemudian, pada praproses juga dilakukan *rescale*. Hal ini dilakukan untuk mempermudah proses pelatihan data.

Tabel 1: Kombinasi model dari *hyperparameter batch size* dan *learning rate*

Kombinasi	<i>Batch Size</i>	<i>Learning Rate</i>
Model 1	8	0.1
Model 2	8	0.01
Model 3	8	0.001
Model 4	8	0.0001
Model 5	8	0.00001
Model 6	16	0.1
Model 7	16	0.01
Model 8	16	0.001
Model 9	16	0.0001
Model 10	16	0.00001
Model 11	32	0.1
Model 12	32	0.01
Model 13	32	0.001
Model 14	32	0.0001
Model 15	32	0.00001

d. Augmentasi

Proses augmentasi gambar penting dilakukan untuk meningkatkan variasi data dan memperkaya set data yang tersedia sehingga model dapat belajar dari berbagai variasi gambar dan lebih tangguh terhadap *overfitting*. Dalam augmentasi ini, beberapa transformasi diterapkan, seperti penyesuaian kecerahan antara 80% hingga 120% dari kecerahan aslinya, rotasi hingga 20 derajat, pergeseran sebesar 0,2 yang menghasilkan efek miring pada gambar, *zoom range* sebesar 0,2, serta *flip* vertikal dan horizontal. Transformasi-transformasi ini membantu meningkatkan ketahanan model terhadap perubahan dan distorsi pada data gambar yang mungkin ditemui pada dunia nyata. Selain itu, mode pengisian *nearest* diterapkan, di mana *pixel* terdekat digunakan untuk mengisi pixel yang kosong akibat transformasi. Transformasi dalam augmentasi ini merujuk pada metode yang umum diterapkan dalam literatur dan praktik terbaik di bidang pengolahan citra dan pembelajaran mesin (Heaton, 2018).

e. Klasifikasi dan Evaluasi

Tahapan klasifikasi diawali dengan ekstraksi fitur menggunakan CNN dengan arsitektur MobileNetV2 menggunakan teknik *transfer learning* untuk mengatasi keterbatasan data (Beikmohammadi dan Faez, 2018). Ekstraksi fitur adalah proses untuk mengambil informasi penting dari data mentah, seperti gambar, yang akan digunakan untuk melatih model. Dalam hal ini, fitur-fitur penting dari citra jerawat diambil oleh MobileNetV2. Setelah ekstraksi fitur, dilakukan *fine-tuning* dengan menambahkan *fully-connected layer* menggunakan fungsi aktivasi ReLU untuk melatih model lebih lanjut pada data jerawat. Pada tahap *fine-tuning*, bobot model *pre-trained* dibekukan hingga *layer* ke-100, memungkinkan *layer-layer* awal untuk mempertahankan pengetahuan yang sudah diperoleh dari set data awal dan hanya melatih *layer-layer* setelahnya.

Dalam langkah mengantisipasi *overfitting* pada optimasi *Adam*, digunakan *earlystopping*, yang merupakan teknik untuk menghentikan pelatihan jika evaluasi model pada data validasi tidak meningkat secara signifikan setelah beberapa *epochs* berturut-turut. *Earlystopping* bekerja dengan memantau metrik performa (seperti *loss* atau akurasi) pada data validasi, dan jika tidak ada peningkatan dalam jumlah *epochs* yang ditentukan (dalam hal ini, 200 *epochs*), pelatihan akan dihentikan lebih awal untuk mencegah *overfitting*. Langkah selanjutnya adalah klasifikasi jerawat ke dalam 4 tingkat keparahan (Level 0 – 3) menggunakan *Global Average Pooling layer* dan *dense* dengan aktivasi *softmax* pada Persamaan 2. *Global Average Pooling* dipilih karena lebih cepat dilatih dan menghindari *overfitting* dibandingkan *fully connected layer* (Kang et al., 2017).

$$softmax(a_i) = \frac{e^{a_i}}{\sum_{j=1}^k e^{a_j}} \tag{2}$$

dengan k = jumlah kelas dan a_i = nilai vektor ke-i.

Kemudian, evaluasi model dilakukan dengan menggunakan *confusion matrix* dan *classification report* pada data uji. *Confusion matrix* memberikan gambaran performa model dengan menggambarkan prediksi benar dan salah untuk setiap kelas (Provost dan Kohavi, 1998), sementara *classification report* memberikan metrik seperti *precision*, *recall*, dan *F1-score* untuk setiap kelas. Akurasi keseluruhan model dihitung dengan membagi jumlah prediksi benar dengan total data uji seperti pada Persamaan 3. Kemudian, berikut ditampilkan *confusion matrix 4x4* pada **Error! Reference source not found.** serta rumus untuk mendapatkan nilai *precision*, *recall*, dan *F1-score* untuk setiap kelas pada Persamaan (4) hingga (15).

Tabel 2: Tampilan *confusion matrix 4x4*

Kelas Aktual	Kelas Prediksi			
	A	B	C	D
A	TP_{AA}	FN_{BA}	FN_{CA}	FN_{DA}
B	FP_{AB}	TP_{BB}	FN_{CB}	FN_{DB}
C	FP_{AC}	FP_{BC}	TP_{CC}	FN_{DC}
D	FP_{AD}	FP_{BD}	FP_{CD}	TP_{DD}

$$Akurasi = \frac{TP_{AA} + TP_{BB} + TP_{CC} + TP_{DD}}{TP_{AA} + FN_{BA} + FN_{CA} + FN_{DA} + FP_{AB} + TP_{BB} + FN_{CB} + FN_{DB} + FP_{AC} + FP_{BC} + FP_{AC} + FP_{BC} + TP_{CC} + FN_{DC} + FP_{AD} + FP_{BD} + FP_{CD} + TP_{DD}} \tag{3}$$

$$Presisi_A = \frac{TP_{AA}}{TP_{AA} + FP_{AB} + FP_{AC} + FP_{AD}} \tag{4}$$

$$Presisi_B = \frac{TP_{BB}}{TP_{BB} + FN_{BA} + FP_{BC} + FP_{BD}} \tag{5}$$

$$Presisi_C = \frac{TP_{CC}}{TP_{CC} + FN_{CA} + FN_{CB} + FP_{CD}} \tag{6}$$

$$\text{Presisi}_D = \frac{TP_{DD}}{TP_{DD} + FN_{DA} + FN_{DB} + FN_{DC}} \quad (7)$$

$$\text{Recall}_A = \frac{TP_{AA}}{TP_{AA} + FN_{BA} + FN_{CA} + FN_{DA}} \quad (8)$$

$$\text{Recall}_B = \frac{TP_{BB}}{TP_{BB} + FP_{AB} + FN_{CB} + FN_{DB}} \quad (9)$$

$$\text{Recall}_C = \frac{TP_{CC}}{TP_{CC} + FP_{AC} + FP_{BC} + FN_{DC}} \quad (10)$$

$$\text{Recall}_D = \frac{TP_{DD}}{TP_{DD} + FP_{AD} + FP_{BD} + FP_{CD}} \quad (11)$$

$$F1 - \text{Score}_A = 2 \times \frac{\text{Presisi}_A \times \text{Recall}_A}{\text{Presisi}_A + \text{Recall}_A} \quad (12)$$

$$F1 - \text{Score}_B = 2 \times \frac{\text{Presisi}_B \times \text{Recall}_B}{\text{Presisi}_B + \text{Recall}_B} \quad (13)$$

$$F1 - \text{Score}_C = 2 \times \frac{\text{Presisi}_C \times \text{Recall}_C}{\text{Presisi}_C + \text{Recall}_C} \quad (14)$$

$$F1 - \text{Score}_D = 2 \times \frac{\text{Presisi}_D \times \text{Recall}_D}{\text{Presisi}_D + \text{Recall}_D} \quad (15)$$

Keterangan:

TP_{ii} = True Positive (Prediksi benar untuk kelas i)

FN_{ij} = False Negative (Kelas aktual i yang salah diprediksi sebagai kelas j)

FP_{ji} = False Positive (Kelas j yang salah diprediksi sebagai kelas i)

i adalah indeks baris (kelas aktual), yaitu A, B, C, dan D

j adalah indeks kolom (kelas prediksi), yaitu juga A, B, C, dan D

Jika $i = j$ (AA, BB, CC, DD) menunjukkan prediksi benar (TP)

Jika $i \neq j$ menunjukkan kesalahan klasifikasi (FP atau FN)

3. Hasil dan Pembahasan

3.1 Pembagian Data

Pada tahap pembagian data, 1457 citra jerawat dibagi menjadi 80% untuk data latih dan 20% untuk data uji. Rincian jumlah data latih dan data uji untuk setiap level

ditampilkan pada Tabel 3. Melalui Tabel 3, terlihat bahwa jumlah masing-masing data latih dan data uji untuk masing-masing level pada penelitian ini terbilang sedikit. Hal ini dapat menyebabkan terjadinya *overfitting*, dimana pada proses pelatihan menghasilkan akurasi yang tinggi, namun akurasi pada proses pengujian sangat rendah (Yudianto *et al.*, 2020). Oleh karena itu, untuk meningkatkan variasi data, perlu dilakukan augmentasi data pada tahap berikutnya.

Tabel 3: Jumlah data latih dan data uji

Level	Jumlah	
	Data Latih	Data Uji
0	398	99
1	510	127
2	149	37
3	110	27

3.2 Praproses Data

Pada tahap praproses data, terdapat beberapa proses penting dilakukan untuk mempersiapkan data gambar sebelum digunakan untuk pelatihan model. Berikut tahapan yang dilakukan pada tahap ini.

- a. Menentukan ukuran gambar pada penelitian ini menyesuaikan dengan arsitektur MobileNetV2 yang digunakan, yaitu sebesar 224×224 *pixel*. Pemilihan ukuran gambar yang sesuai sangat penting karena dapat memengaruhi keakuratan model. Semakin kecil ukuran gambar, semakin kurang akurat hasilnya, tetapi juga semakin cepat waktu pelatihan dan komputasinya (Rochmawanti *et al.*, 2021). Hasil praproses data dengan mengubah ukuran gambar sebesar 224×224 *pixel* ditunjukkan pada Gambar 4.



(a) Data citra asli



(b) Data citra hasil praproses

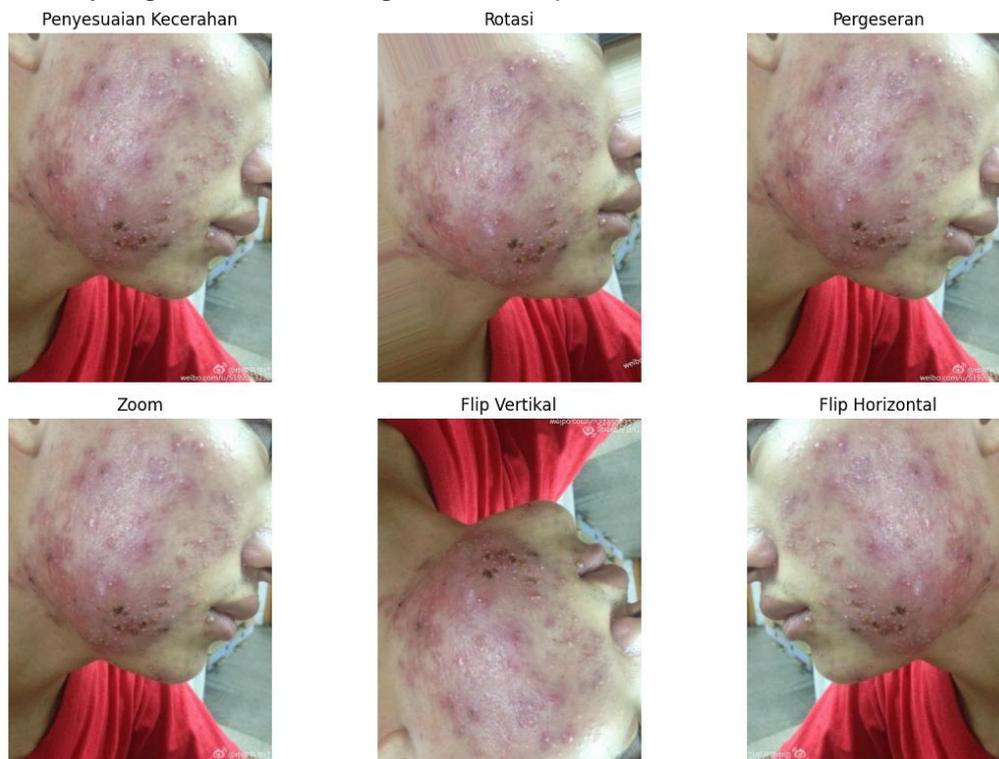
Gambar 4: Hasil praproses (a) data citra asli dan (b) data citra hasil praproses

- b. Menentukan *hyperparameter* seperti *batch size* dan *learning rate* yang akan digunakan. Penelitian ini menggunakan *batch size* sebesar 8, 16, dan 32 serta *learning rate* sebesar 0,1, 0,01, 0,001, 0,0001, 0,00001. Kemudian, dari nilai-nilai ini akan membentuk 15 model yang berbeda.

- c. Menentukan *epochs* yang akan digunakan. Pada penelitian ini, ditentukan dengan batas maksimal sebanyak 200 *epochs*.
- d. Menentukan optimasi yang akan digunakan untuk meminimalkan fungsi kerugian (*loss function*) selama pelatihan model. Pada penelitian ini menggunakan optimasi *Adam*.
- e. Menentukan jumlah kelas yang akan digunakan. Pada penelitian ini terdapat 4 jumlah kelas, yang berarti model akan dilatih untuk mengklasifikasikan gambar ke dalam 4 kelas yang berbeda, yaitu level 0, level 1, level 2, dan level 3.
- f. Melakukan *rescaling* dengan membagi setiap nilai *pixel* dalam saluran warna (seperti merah, hijau, dan biru) dengan 255, yang merupakan nilai maksimum dalam representasi gambar digital, memungkinkan nilai *pixel* untuk berada dalam rentang 0 – 1. Misalnya, jika nilai *pixel* asli adalah 128, setelah *rescale* akan menjadi $\frac{128}{255} = 0,5$. Proses ini bertujuan untuk mempermudah proses pelatihan data dan meningkatkan stabilitas numerik.

3.3 Augmentasi

Setelah dilakukan praproses data, penelitian ini juga menerapkan teknik augmentasi dalam upaya memperkaya keragaman data yang digunakan melalui transformasi atau modifikasi pada citra-citra asli (Shorten dan Khoshgoftaar, 2019). Pada tahap ini transformasi yang dilakukan, yaitu penyesuaian kecerahan 80% hingga 120% dari kecerahan gambar asli, rotasi hingga 20 derajat, pergeseran sebesar 0.2 yang menghasilkan efek miring pada gambar, zoom range sebesar 0.2 serta *flip* vertikal dan horizontal. Kemudian, dilakukan mode pengisian *nearest*, yaitu *pixel* terdekat yang digunakan untuk mengisi *pixel* yang kosong. Berikut ditampilkan hasil transformasi yang dilakukan dari gambar asli pada Gambar 5.



Gambar 5: Hasil transformasi dari proses augmentasi

Langkah ini penting untuk meningkatkan variasi data pelatihan yang dapat memengaruhi kinerja model. Kemudian, rasionalisasi dari setiap transformasi dapat dilihat langsung dari Gambar 5. Penyesuaian kecerahan mensimulasikan variasi pencahayaan, rotasi dan pergeseran membantu model mengenali kondisi kulit dari berbagai sudut dan posisi, *zoom* melatih pengenalan pada skala berbeda, sedangkan *flip* menciptakan perspektif baru. Teknik-teknik ini meningkatkan kemampuan model untuk mengenali fitur-fitur penting kondisi kulit terlepas dari variasi dalam pengambilan gambar sehingga meningkatkan *robustness* dan generalisasi model (Wang dan Perez, 2017). Dengan memperkaya set data melalui augmentasi, model diharapkan dapat lebih akurat dalam menilai tingkat keparahan kondisi kulit dari berbagai jenis input, mengurangi *overfitting*, dan meningkatkan performa secara keseluruhan (Mikolajczyk dan Grochowski, 2018).

Setelah dilakukan tahapan augmentasi, jumlah variasi data pelatihan dari setiap level menjadi lebih banyak. Perlu ditekankan bahwa proses augmentasi hanya diterapkan pada data latih, sedangkan data uji tetap dalam kondisi asli. Pendekatan ini menjamin evaluasi model yang objektif dan tidak bias. Perbandingan jumlah data latih serta data uji sebelum dan setelah dilakukan augmentasi ditampilkan pada Tabel 4.

Tabel 4: Jumlah data latih dan data uji sebelum dan setelah augmentasi

Level	Jumlah Sebelum Augmentasi		Jumlah Setelah Augmentasi	
	Data Latih	Data Uji	Data Latih	Data Uji
0	398	99	497	99
1	510	127	637	127
2	149	37	186	37
3	110	27	137	27

3.4 Klasifikasi dan Evaluasi

Setelah dilakukan pembagian data, praproses data, dan augmentasi data, akan dibangun 15 model CNN menggunakan teknik *transfer learning*. Model CNN yang digunakan didasarkan pada model MobileNetV2. Lima belas model yang dibangun dari penelitian ini akan dibandingkan nilai akurasi, *precision*, *recall*, dan *F1-score* untuk mendapatkan model CNN dengan performa terbaik. Sebanyak 15 model yang akan dibangun didasarkan atas kombinasi *batch size* dan *learning rate* serta jumlah *epochs* yang dibatasi hingga 200 pada saat pelatihan. Hasil evaluasi model ditampilkan pada Tabel 5.

Tabel 5: Evaluasi Model 1 sampai 15

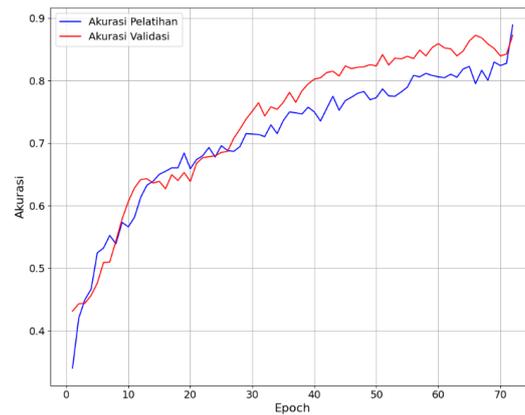
Model	<i>Epochs</i>	Akurasi	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Model 1	9	43.75%	11%	25%	15%
Model 2	10	43.45%	11%	25%	15%
Model 3	7	65.49%	55%	52%	52%
Model 4	16	68.94%	79%	60%	62%
Model 5	61	86.43%	89%	85%	86%
Model 6	8	43.75%	11%	25%	15%
Model 7	8	43.43%	11%	25%	15%
Model 8	11	73.93%	70%	74%	70%
Model 9	6	40.63%	18%	28%	20%

Model	Epochs	Akurasi	Precision	Recall	F1-Score
Model 10	71	87.29%	89%	84%	86%
Model 11	6	44.47%	11%	25%	15%
Model 12	11	36.21%	13%	26%	15%
Model 13	9	51.41%	63%	34%	31%
Model 14	6	49.30%	25%	31%	27%
Model 15	57	78.97%	83%	72%	75%

Tahap klasifikasi melibatkan pengujian model klasifikasi menggunakan data uji untuk mengevaluasi seberapa akurat model tersebut dapat mengklasifikasikan data ke dalam kelas atau kategori yang tepat. Kinerja setiap model akan dievaluasi dengan menghitung tingkat keakuratan prediksi atau klasifikasi yang dihasilkan oleh model tersebut. Berdasarkan hasil rata-rata evaluasi model yang terdapat pada **Error! Reference source not found.**, terlihat bahwa terdapat variasi nilai akurasi serta rata-rata dari presisi, *recall*, dan *F1-score* yang signifikan antar model. Hal ini mengindikasikan bahwa pemilihan *hyperparameter* seperti *learning rate* memiliki pengaruh besar terhadap kinerja model klasifikasi (Rochmawanti et al., 2021).

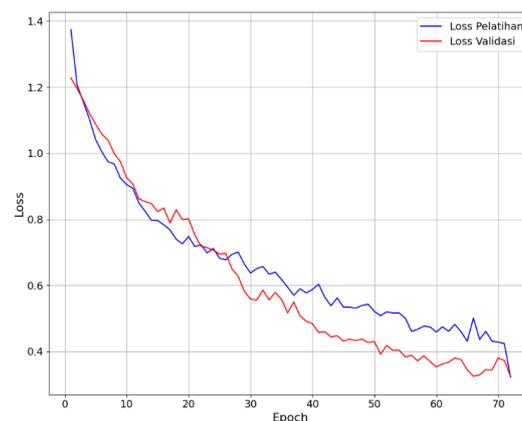
Model dengan akurasi tertinggi adalah Model 10 sebesar 87,29%, diikuti oleh Model 5 sebesar 86,43%, dan Model 15 sebesar 78,97%. Kinerja model ini lebih baik dibandingkan model lainnya dalam mengklasifikasikan data uji dengan benar. Sementara itu, pada Model 9 (40,63%) dan Model 12 (36,21%) memiliki akurasi yang relatif rendah. Hal ini dapat disebabkan oleh *overfitting* atau *underfitting* selama proses pelatihan, yang mungkin terkait dengan pemilihan *hyperparameter* yang kurang baik.

Selain akurasi, metrik lain seperti rata-rata presisi, *recall*, dan *F1-score* juga penting untuk dievaluasi. *Precision* mengukur proporsi sampel yang diprediksi sebagai kelas positif (misalnya, level 1) yang benar-benar positif. Sementara itu, *recall* mengukur sejauh mana model berhasil mengidentifikasi sampel positif dengan benar. Model 10 memiliki nilai rata-rata presisi yang tinggi sebesar 89%. Hal ini menunjukkan bahwa dari seluruh sampel yang diprediksi sebagai kelas positif, sebagian besar (89%) benar-benar positif. Artinya, Model 10 memiliki tingkat kepercayaan yang tinggi dalam memprediksi kelas positif. Sementara itu, Model 10 juga memiliki nilai rata-rata *recall* yang tinggi sebesar 84%, artinya sebagian besar (84%) sampel positif yang ada berhasil diidentifikasi dengan benar oleh model sebagai kelas positif. Nilai rata-rata *F1-score* pada Model 10 juga tinggi, yaitu 86%. Hal ini menunjukkan keseimbangan yang baik antara rata-rata presisi dan *recall*. Kombinasi *batch size* 16 dan *learning rate* 0.00001 pada Model 10 tampaknya merupakan kombinasi *hyperparameter* terbaik. Jumlah *epochs* sebanyak 71 juga memungkinkan model untuk mencapai kinerja terbaik dengan rata-rata akurasi 87,29%. Oleh karena itu, berdasarkan informasi pada **Error! Reference source not found.**, Model 10 menghasilkan performa terbaik untuk mengklasifikasi tingkat keparahan jerawat dibandingkan model lainnya. Kemudian, untuk memvisualisasikan kinerja Model 10 selama proses pelatihan dengan menggunakan *loss function categorical crossentropy* yang mengukur perbedaan antara distribusi probabilitas prediksi model dengan distribusi sebenarnya dari kelas dalam klasifikasi multi-kelas (Heaton, 2018). Berikut ditampilkan plot akurasi dan *loss categorical crossentropy* pada Gambar 6 dan Gambar 7.



Gambar 6: Perbandingan nilai akurasi pelatihan dan pengujian

Gambar 6 menunjukkan plot akurasi Model 10 selama proses pelatihan. Akurasi data pelatihan ditandai dengan garis biru dan akurasi data pengujian ditandai dengan garis merah. Terlihat bahwa akurasi pada data pelatihan dan pengujian meningkat bersamaan dengan bertambahnya *epochs* hingga mencapai akurasi validasi 87% di akhir pelatihan. Hal ini sejalan dengan hasil evaluasi terbaik pada Tabel 5. Sementara itu, akurasi pelatihan dan pengujian tidak terdapat *gap* yang besar, artinya model tidak mengalami *overfitting*.



Gambar 7: Perbandingan nilai *loss categorical crossentropy* pelatihan dan pengujian

Gambar 7 menunjukkan plot *loss categorical crossentropy*, dengan garis biru untuk data pelatihan dan garis merah untuk data pengujian yang terus menurun seiring bertambahnya *epochs*. *Gap* antara *loss categorical crossentropy* pelatihan dan pengujian tidak besar di akhir pelatihan, artinya model dapat meminimalkan *error* prediksi tanpa *overfitting*, dengan *loss* validasi mencapai 0,32. Visualisasi ini menunjukkan bahwa Model 10 memiliki kinerja yang baik, sejalan dengan hasil evaluasi pada **Error! Reference source not found.**

4. Simpulan dan Saran

Penelitian ini telah mengaplikasikan model CNN MobileNetV2 dengan teknik *transfer learning* untuk mengklasifikasikan tingkat keparahan jerawat berdasarkan set data ACNE04. Setelah melakukan evaluasi dengan berbagai kombinasi *hyperparameter*, diperoleh hasil terbaik pada Model 10 dengan *batch size* 16 dan *learning rate* 0,00001. Model ini mencapai kinerja yang baik dengan akurasi sebesar 87,29%, *precision*

sebesar 89%, *recall* sebesar 84%, dan *F1-score* sebesar 86%. Kombinasi *hyperparameter* yang digunakan dalam penelitian ini memungkinkan model belajar secara efektif dari data dan mencapai keseimbangan yang baik antara *precision* dan *recall* dalam mengklasifikasikan tingkat keparahan jerawat. Penggunaan *batch size* yang kecil (16) memungkinkan model untuk melakukan *update* parameter lebih sering, sementara *learning rate* yang rendah (0,00001) membantu model melakukan *fine-tuning* yang lebih halus, yang sangat bermanfaat dalam konteks *transfer learning*. Penggunaan *transfer learning* dengan model CNN MobileNetV2 dalam klasifikasi tingkat keparahan jerawat, memberikan kontribusi penting dalam pengembangan metode diagnosis yang lebih akurat.

Meskipun hasil yang diperoleh sudah cukup baik, masih ada ruang untuk pengembangan lebih lanjut dalam upaya meningkatkan kinerja model klasifikasi tingkat keparahan jerawat dengan menerapkan teknik augmentasi data yang lebih beragam, seperti rotasi, pencerminan, pemutaran, dan perubahan skala pada data gambar. Hal tersebut dapat membantu meningkatkan keragaman data dan mencegah *overfitting*. Selain itu, mengeksplorasi arsitektur CNN lainnya yang lebih sesuai dengan karakteristik data citra jerawat juga dapat dilakukan. Arsitektur seperti EfficientNet, DenseNet, atau InceptionNet berpotensi memberikan kinerja yang lebih baik. Optimalisasi *hyperparameter* secara ekstensif juga dapat dilakukan dengan mencoba rentang nilai yang lebih luas atau menggunakan teknik optimasi seperti *grid search* atau *random search*. Memasukkan metode klasifikasi tambahan seperti *ensemble learning* yang menggabungkan beberapa model untuk meningkatkan akurasi, juga dapat dipertimbangkan.

Daftar Pustaka

- Achmad, Y. F., Yulfitri, A., & Ulum, M. B. (2021). Identifikasi Jenis Jerawat Berdasarkan Tekstur Menggunakan GLCM dan Backpropagation. *Jurnal Sains Manajemen Informatika Dan Komputer*, 20(2), 139–146.
- Albertino, T. (2023). Klasifikasi citra hispatologi kanker payudara dengan metode CNN dan transfer learning [Skripsi]. Bogor(ID): Institut Pertanian Bogor.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 2–74.
- Becker, M., Wild, T., & Zouboulis, C. C. (2017). Objective assessment of acne. *Clinics in Dermatology*, 35(2), 147–155.
- Beikmohammadi, A., & Faez, K. (2018). Leaf classification for plant recognition with deep transfer learning. *2018 Irian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 21–26. doi.org/10.1109/ICSPIS.2018.8700547.
- Hadianti, S., Sastypratiwi, H., & Sukamto, S. A. (2015). Sistem pakar diagnosis jenis jerawat pada wajah menggunakan metode k-means clustering. *Jurnal Sistem Dan Teknologi Informasi*, 3(3), 1–5.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. 3rd ed. Cambridge(US): Morgan kaufmann.
- Hasanah, R. L., & Hasan, M. (2022). Deteksi lesi acne vulgaris pada citra jerawat wajah menggunakan metode k-means clustering. *Indonesian Journal on Software Engineering (IJSE)*, 8(1), 46–51.
- Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: deep learning. *Genetic Programming and Evolvable Machines*, 19(1–2), 305–307. doi.org/10.1007/s10710-017-9314-z.

- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., & Andreetto, M. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. doi.org/10.48550/arXiv.1704.04861.
- Jordan, M. I., & Mitchell, T. M. (2015). Data, privacy, and the greater good. *Science*, 349(6245), 253–255. doi.org/10.1126/science.aac4520.
- Kang, G., Liu, K., Hou, B., & Zhang, N. (2017). 3D multi-view convolutional neural networks for lung nodule classification. *PLoS ONE*, 12(11), 1–21. doi.org/10.1371/journal.pone.0188290.
- Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations, San Diego*. doi.org/10.48550/arXiv.1412.6980.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. doi.org/10.1038/nature14539.
- Melisa, Lestari, P., & Irawati. (2022). Sistem pakar pendiagnosa jenis jerawat pada wajah berbasis web menggunakan metode certainty factor. *Buletin Sistem Informasi dan Teknologi Islam (BUSITI)*, 3(1), 79–85.
- Mikolajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *IEEE International Interdisciplinary PhD Workshop (IIPHDW)*, 117–122. doi.org/10.1109/IIPHDW.2018.8388338.
- Okoro, E., Ogunbiyi, A., & George, A. (2016). Prevalence and pattern of acne vulgaris among adolescents in Ibadan, south-west Nigeria. *Journal of the Egyptian Women's Dermatologic Society*, 13(1), 7–12.
- Provost, F., & Kohavi, R. (1998). *On Applied Research in Machine Learning*. *Machine Learning Boston*, 30(1), 1–6.
- Ramadhani, M., M.T, S., & K, H. B. D. (2018). Klasifikasi jenis jerawat berdasarkan tekstur dengan menggunakan metode GLCM. *e-Proceeding of Engineering*, 5(1), 870–876.
- Rianto, R., & Listianto, D. R. (2023). Convolutional Neural Network untuk mengklasifikasi tingkat keparahan jerawat. *AITI: Jurnal Teknologi Informasi*, 20(2), 167–176.
- Rochmawanti, O., Utamingrum, F., & Bachtiar, F. A. (2021). Analisis performa pre-trained model convolutional neural network dalam mendeteksi penyakit tuberculosis. *Jurnal Teknologi Infomasi dan Ilmu Komputer (JTIK)*, 8(4), 805-814.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: inverted residuals and linear bottlenecks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520.
- Sellat, Q., Bisoy, S. K., & Priyadarshini, R. (2022). Cognitive big data intelligence with a metaheuristic approach. Di dalam: Mishra S, Tripathy HK, Mallick PK, Sangaiah AK, Chae GS, editor. *Cognitive Data Science in Sustainable Computing*. Cambridge (AS): Academic Press, hlm 211–238.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(60), 1–48. doi.org/10.1186/s40537-019-0197-0.
- Sibero, H. T., Sirajudin, A., & Anggraini, D. I. (2019). Prevalensi dan gambaran epidemiologi akne vulgaris di provinsi lampung. *Jurnal Kedokteran UNILA*, 3(2), 309–312.
- Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. doi.org/10.48550/arXiv.1712.04621.

- Wu, X., En, N., Liang, J., Lai, Y.-K., She, D., Cheng, M.-M., & Yang, J. (2019). Joint acne image grading and counting via label distribution learning. *IEEE International Conference on Computer Vision (ICCV)*, 10641–10650.
- Yudianto, M. R. A., Kusriani, & Al Fatta, H. (2020). Analisis pengaruh tingkat akurasi klasifikasi citra wayang dengan algoritma convolutional neural network, 4(2), 182–190.