

Klasifikasi Jenis Jerawat Secara Otomatis Dengan *Convolutional Neural Network* Menggunakan Arsitektur Resnet-50

Automatic Acne Type Classification with Convolutional Neural Network Using Resnet-50 Architecture

Muhammad Iqbal Anshori¹, Rifki Ainul Yaqqin², Muhammad Ali Zafar Sidiq³, Ignatius Wiseto Prasetyo Agung⁴

Program Studi Teknik Informatika, Universitas Adhirajasa Reswara Sanjaya (ARS University), Bandung, Indonesia

*E-mail: iqbalanshr@gmail.com

Abstrak

Jerawat adalah masalah kulit umum yang memerlukan penanganan berbeda berdasarkan jenisnya, seperti komedo, konglobata, dan papulopustular. Penelitian ini mengembangkan sistem klasifikasi jenis jerawat otomatis menggunakan arsitektur Residual Network (ResNet-50) berbasis deep learning. Dengan 50 lapisannya, ResNet-50 efektif dalam klasifikasi gambar. Tujuan penelitian ini adalah mengklasifikasikan jenis jerawat dari gambar kulit pada wajah, sehingga dapat membantu diagnosis dan perawatan. Metode yang digunakan dalam penelitian ini mencakup beberapa tahap utama, yaitu pengumpulan dataset, pelatihan model menggunakan CNN dengan arsitektur ResNet-50, pengujian model, serta evaluasi kinerja. Dataset diperoleh dari Roboflow, terdiri dari tiga kelas: acne-comedonica, acne-conglobata, dan acne-papulopustulosa. Proses melibatkan preprocessing gambar, augmentasi data, dan penyesuaian parameter model, termasuk teknik dropout dan optimizer Adam. Model dapat mencapai akurasi 98.35% dengan loss 0.0489 dan akurasi validasi tertinggi sebesar 92.86% dengan loss validasi 0.1976. Selain itu, analisis confusion matrix menunjukkan hasil akurasi 93%, yang mengindikasikan kinerja model dalam membedakan antara kelas-kelas jerawat secara efektif. Hasil ini menunjukkan bahwa model efektif dalam mengklasifikasikan jenis jerawat dan dapat memberikan dampak signifikan dalam membantu diagnosis yang lebih akurat dan cepat.

Kata kunci: Convolutional Neural Network, Resnet-50, Deep Learning, Klasifikasi Gambar.

Abstract

Acne is a common skin problem that requires different treatments based on its type, such as blackheads, conglobata, and papulopustular. This research develops an automatic acne type classification system using deep learning-based Residual Network (ResNet-50) architecture. With its 50 layers, ResNet-50 is effective in image classification. The objective of this research is to classify the type of acne from skin images on the face, so that it can help diagnosis and treatment. The method used in this research includes several main stages, namely the collection of the dataset, model training using CNN with ResNet-50 architecture, model testing, and performance evaluation. The dataset was obtained from Roboflow, consisting of three classes: acne-comedonica, acne-conglobata, and acne-papulopustulosa. The process involves image preprocessing, data augmentation, and model parameter adjustment, including Adam's dropout and optimizer techniques. The model can achieve 98.35% accuracy with loss of 0.0489 and the highest validation accuracy of 92.86% with a validation loss of 0.1976. In addition, confusion matrix analysis shows an accuracy result of 93%, which indicates the performance of the model in distinguishing between acne classes effectively. These results show that the model is effective in classifying the types of acne and can have a significant impact in assisting a more accurate and faster diagnosis.

Keywords: Convolutional Neural Network, Resnet-50, Deep Learning, Image Classification

Naskah diterima 06 Ags. 2024; direvisi 26 Sept. 2024; dipublikasikan 01 Apr. 2025.
JAMIKA is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



I. PENDAHULUAN

Jerawat adalah masalah peradangan kronis yang terjadi di unit folikel rambut dan kelenjar minyak, sering sembuh sendiri, dan dipicu oleh bakteri *Cutibacterium acnes* selama masa remaja dengan pengaruh hormon dehydroepiandrosterone (DHEA). Penyakit ini umum terjadi, mempengaruhi 80%-100% populasi, dan ditandai oleh lesi peradangan dan non-peradangan di wajah, lengan atas, dada, dan punggung [1]. Selain pada remaja, jerawat merupakan salah satu masalah kulit yang sering terjadi pada banyak orang dewasa juga.

Ada berbagai jenis jerawat yang memiliki karakteristik tersendiri dan metode penanganan yang berbeda-beda contohnya seperti komedo, konglobata dan papulopustular[2], [3].

Jerawat berdampak tidak hanya secara fisik, tetapi juga secara psikologis, dengan salah satu efeknya adalah depresi. Jerawat dapat mempengaruhi kesehatan mental, termasuk masalah psikologis seperti rasa malu, beban emosional, keengganan untuk memakai pakaian yang menonjolkan area jerawat, kemarahan, serta menurunnya rasa percaya diri dan perasaan distigmatisasi. Dampak psikiatris seperti depresi, kecemasan, dan bahkan pemikiran untuk bunuh diri juga sering dikaitkan dengan kondisi ini. Meningkatnya masalah psikologis, psikiatris, dan fisik pada penderita jerawat dalam jangka panjang dapat memperberat beban penyakit, sehingga berdampak negatif pada kualitas hidup[4].

Pada remaja, stres yang tinggi menjadi faktor risiko signifikan dalam berkembangnya jerawat. Perasaan tertekan yang sering dialami oleh remaja, yang sering disebut stres, dapat mengakibatkan akumulasi gangguan psikologis serta kondisi fisik yang memicu dan memperburuk jerawat. Stres psikologis ini merangsang hipotalamus untuk menghasilkan Hormon Pelepasan *Corticotropin* (CRH), yang kemudian memicu peningkatan produksi hormon androgen seperti Hormon *Adrenocorticotropin* (ACTH). Hormon ini berperan penting dalam memunculkan dan memperparah jerawat, menunjukkan hubungan kuat antara kondisi psikologis dan keparahan jerawat pada kelompok usia remaja[5].

Oleh karena itu, pengklasifikasian jenis jerawat secara otomatis menjadi langkah yang sangat penting untuk membantu menentukan perawatan yang lebih cepat dan tepat. Dengan kemajuan teknologi, sistem berbasis kecerdasan buatan dapat digunakan untuk menganalisis dan mengidentifikasi berbagai jenis jerawat dengan akurasi tinggi. Hal ini tidak hanya mempercepat diagnosis tetapi juga memungkinkan penanganan yang lebih spesifik sesuai dengan jenis jerawat yang dialami. Dengan metode ini, baik pasien maupun tenaga medis dapat lebih efisien dalam menentukan strategi perawatan, sehingga potensi dampak fisik dan psikologis dari jerawat dapat diminimalkan lebih awal.

Kecerdasan buatan telah mengalami perkembangan pesat dan kini mencakup berbagai cabang, seperti *Machine Learning* dan *Deep Learning*, yang telah diterapkan dalam berbagai sektor. Kecerdasan buatan tidak hanya digunakan untuk otomatisasi tugas-tugas sederhana, tetapi juga telah menjadi komponen penting dalam bidang-bidang kompleks seperti pengenalan wajah, pemrosesan bahasa alami, dan bahkan dalam bidang medis untuk mendiagnosis penyakit. Dalam konteks analisis citra medis, kecerdasan buatan telah memberikan kontribusi signifikan dengan meningkatkan akurasi diagnosis serta mempercepat proses identifikasi berbagai kondisi, termasuk masalah kulit seperti jerawat[6].

Deep learning merupakan salah satu teknologi kecerdasan buatan yang dapat mengklasifikasikan suatu gejala, termasuk jerawat dengan otomatis. *Deep Learning* adalah cabang dari *Machine Learning* yang menggunakan Jaringan Syaraf Tiruan (JST), atau versi lanjutan dari JST, untuk memungkinkan komputer berpikir dan bertindak mirip dengan cara manusia. *Deep Learning* diterapkan pada berbagai jenis data seperti gambar, suara, teks, dan video. Dalam metode ini, model dilatih dengan dataset besar yang memiliki label, yang kemudian diubah menjadi representasi internal berupa nilai piksel atau *feature vector*. Representasi ini digunakan untuk mendeteksi dan mengklasifikasikan pola dalam data input[7].

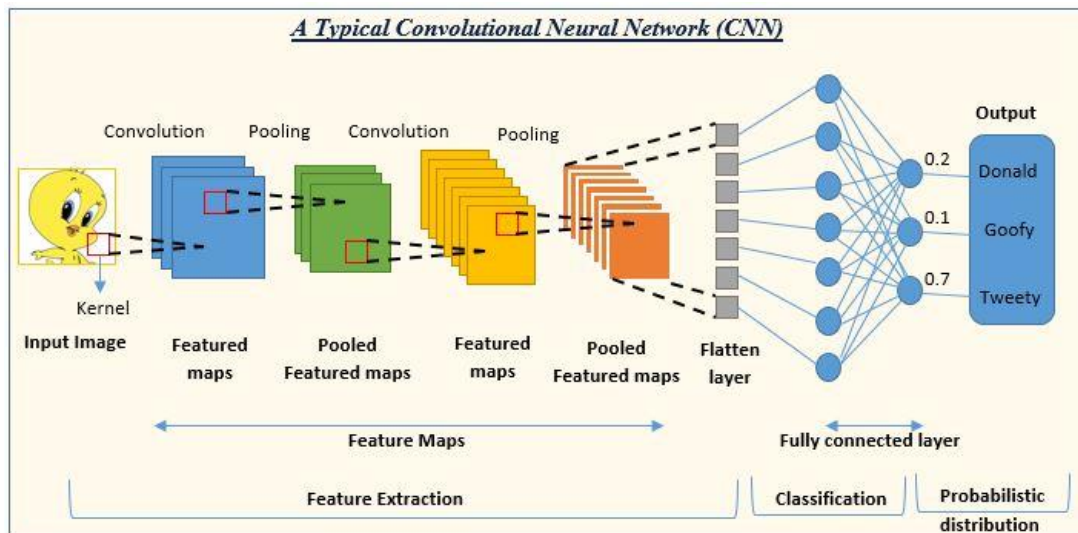
Salah satu yang Algoritma paling efektif dalam *deep learning* untuk pengolahan data citra adalah *Convolutional Neural Network* (CNN) yang merupakan bagian integral dari jaringan saraf tiruan dalam *deep neural network*, dirancang khusus untuk mengenali dan memproses data visual seperti gambar. Algoritma ini meniru cara kerja visual cortex pada otak manusia dan hewan, yang bertanggung jawab dalam pemrosesan rangsangan visual, sehingga membuat CNN unggul dalam tugas pengenalan dan klasifikasi citra dibandingkan dengan algoritma *neural network* lainnya[8].

CNN terdiri dari beberapa lapisan, termasuk *input layer*, *convolutional layer*, *activation layer*, *pooling layer*, dan *fully connected layer*. Lapisan *convolutional layer* menggunakan filter untuk mengaplikasikan kernel pada citra, menghasilkan output yang berisi fitur-fitur spasial dari citra. Lapisan *fully connected layer* kemudian menghubungkan fitur-fitur tersebut untuk menghasilkan output akhir berupa klasifikasi citra[9], [10].

Seperti yang sudah dijelaskan, pada Gambar 1 dapat dilihat, secara garis besar arsitektur CNN terbagi menjadi dua bagian utama, yaitu *Feature Extraction Layer* (FEL) dan *Fully-Connected Layer* (FCL). FEL ini bertugas meng-encode gambar menjadi fitur-fitur yang berupa angka-angka yang merepresentasikan gambar tersebut. FEL terdiri dari dua bagian, yaitu *Convolutional Layer* dan *Pooling Layer*[11].

Convolutional Layer adalah komponen kunci dalam metode CNN, menghasilkan citra baru yang menampilkan fitur dari citra input. Layer ini menggunakan filter dan terdiri dari neuron yang membentuk filter berukuran pixel. Filter ini digeser ke seluruh bagian gambar, dan setiap pergeseran melakukan operasi “dot” antara input dan nilai filter, menghasilkan output yang dikenal sebagai *activation map* atau *feature map* [12].

Selanjutnya, *Pooling Layer* biasanya terletak setelah lapisan konvolusional. Lapisan ini menggunakan filter dengan ukuran tertentu yang digeser ke seluruh area *feature map*. Penggabungan yang umum digunakan adalah *Max Pooling*, yang membagi output dari lapisan konvolusional menjadi beberapa grid kecil dan mengambil nilai maksimum dari setiap grid untuk menyusun matriks citra yang telah direduksi. Tujuannya adalah untuk mengurangi dimensi dari *Feature Map* sehingga mempercepat komputasi [13], [14].



Gambar 1. Arsitektur CNN

FCL merupakan bagian kedua dari arsitektur CNN mengubah dimensi data sehingga data dapat diklasifikasikan secara linear. Alih-alih menggunakan operasi konvolusi, lapisan ini memanfaatkan komputasi perkalian matriks yang ditambahkan dengan *bias offset* untuk menghasilkan *output*. Setiap neuron dalam FCL memiliki koneksi penuh ke semua aktivasi pada lapisan sebelumnya, sehingga dinamakan *fully connected layer*[15].

Dalam bidang pengenalan gambar, arsitektur CNN telah berkembang dengan sangat pesat dan beragam. Banyak variant arsitektur CNN yang telah dikembangkan untuk menangani berbagai tugas pengenalan pola dan klasifikasi gambar. Salah satu varian arsitektur CNN yang sangat baik untuk tugas klasifikasi gambar adalah *Residual Network* (ResNet). ResNet, khususnya ResNet-50 merupakan sebuah struktur jaringan saraf tiruan yang sangat dalam dan telah memberikan dampak besar dalam bidang pengenalan gambar. Dengan 50 lapisan, arsitektur ini telah digunakan secara sukses dalam berbagai aplikasi untuk pengenalan objek dan klasifikasi gambar [16]. ResNet mengatasi masalah hilangnya gradien yang sering terjadi pada jaringan neural yang sangat dalam dengan menggunakan sistem *skip connection*. Sistem ini memungkinkan bagian perhitungan perkalian model dilewati dan informasi langsung masuk ke lapisan berikutnya, memastikan bahwa informasi penting tidak hilang selama proses pelatihan [17].

Penelitian ini menggunakan ResNet-50 karena arsitektur ini cocok untuk tugas klasifikasi gambar. Dengan menggunakan ResNet-50, penelitian ini dapat mengatasi masalah hilangnya gradien yang sering terjadi pada jaringan neural yang sangat dalam. Selain itu, arsitektur ini juga telah terbukti memberikan hasil yang akurat dalam berbagai aplikasi pengenalan objek dan klasifikasi gambar[18]. Oleh karena itu, penggunaan ResNet-50 dalam penelitian ini sangat relevan dan efektif untuk mencapai hasil yang diinginkan.

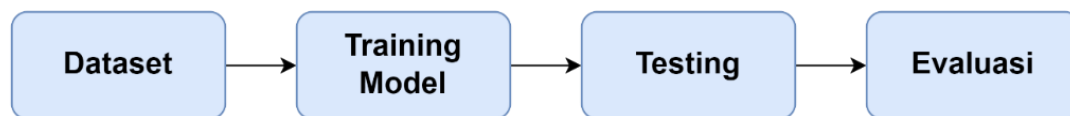
Berdasarkan dua artikel penelitian, yaitu penelitian yang dilakukan oleh Hasan dkk[19]. dan Sudana Putra dkk[20], ditemukan bahwa model Convolutional Neural Network (CNN) memiliki potensi yang signifikan dalam klasifikasi jerawat. Penelitian oleh Hasan dkk. menunjukkan bahwa model CNN dapat mencapai akurasi 91,6% dalam klasifikasi jerawat. Sementara itu, penelitian oleh Sudana Putra dkk. memperoleh hasil akurasi deteksi jerawat yang sangat tinggi, yaitu antara 99.8% hingga 100%. Kedua penelitian ini mengindikasikan bahwa penggunaan CNN dalam aplikasi klinis dapat sangat membantu dalam diagnosis dan penanganan jerawat secara efisien. Oleh karena itu, pengembangan teknologi deep learning dalam bidang kesehatan, khususnya dalam deteksi dan klasifikasi jerawat, sangat menjanjikan dan dapat memberikan manfaat bagi tenaga medis dan pasien.

Penelitian ini memiliki relevansi dengan kedua studi tersebut karena sama-sama menerapkan CNN untuk pengklasifikasian jenis jerawat. Namun, terdapat perbedaan yang signifikan dalam metodologi yang digunakan. Pada penelitian ini, diterapkan arsitektur ResNet-50, yang merupakan varian dari CNN dengan kemampuan lebih mendalam dan lebih efektif dalam mengatasi masalah vanishing gradient[21], dibandingkan dengan arsitektur CNN yang digunakan dalam penelitian Hasan dkk. dan Sudana Putra dkk. Adapun tujuan dari penelitian ini adalah mengklasifikasikan jenis jerawat dari gambar kulit pada wajah untuk membantu diagnosis dan perawatan yang lebih akurat dan cepat.

Penelitian ini akan membahas semua langkah-langkah yang diperlukan untuk mengembangkan sistem klasifikasi jerawat otomatis, termasuk pengumpulan data, pelatihan model Resnet-50, dan evaluasi kinerja model. Diharapkan penelitian ini akan membantu penderita jerawat, bidang dermatologi, dan penggunaan AI dalam perawatan kulit dan memberikan kontribusi pengetahuan dalam penerapan CNN menggunakan arsitektur ResNet-50. Namun, penelitian ini memiliki beberapa keterbatasan, termasuk keterbatasan pada dataset yang mungkin kurang representatif dan penggunaan perangkat keras yang membatasi eksplorasi berbagai parameter. Keterbatasan perangkat keras ini mengakibatkan kesulitan dalam mencoba berbagai konfigurasi model, karena proses pelatihan memerlukan sumber daya yang besar.

II. METODE PENELITIAN

Dalam penelitian ini, seperti yang digambarkan pada Gambar 2, terdapat beberapa langkah utama dimulai dari pengumpulan Dataset yang berisi gambar jerawat, Training Model menggunakan CNN dengan arsitektur ResNet-50, Testing dan terakhir proses Evaluasi.



Gambar 2. Flowchart Metode Penelitian

Dataset

Dataset yang digunakan dalam penelitian ini diperoleh dari Roboflow, sebuah platform yang menyediakan berbagai dataset untuk keperluan penelitian dan pengembangan model berbasis kecerdasan buatan, khususnya dalam bidang computer vision. Roboflow menyediakan dataset yang telah disiapkan dan dioptimalkan untuk berbagai tugas klasifikasi dan deteksi objek, termasuk klasifikasi jenis jerawat. Dataset yang digunakan berfokus pada klasifikasi tiga jenis jerawat, yaitu acne-comedonica, acne-papulopustulosa, dan acne-conglobata yang memiliki ciri dan karakteristik tersendiri. Acne comedonica memiliki ciri seperti titik-titik hitam (*blackheads*) atau putih (*whiteheads*) pada kulit, acne papulopustulosa memiliki ciri benjolan merah dan berisi nanah dan acne conglobata memiliki ciri Nodul besar, abses, dan peradangan luas. Setiap gambar dalam dataset ini memiliki resolusi 640x640 piksel dan berformat JPG, yang memudahkan proses preprocessing dan pelatihan model. (<https://universe.roboflow.com/taschenbier/acne-type-classification>)

Secara keseluruhan, dataset ini terdiri dari 754 gambar yang telah dibagi menjadi dua subset, yaitu 726 gambar untuk pelatihan (training) dan 28 gambar untuk pengujian (testing). Proporsi ini memastikan bahwa model dapat dilatih dengan cukup data untuk mengenali pola yang spesifik dari setiap jenis jerawat, sambil tetap menyisakan data untuk mengukur performa model pada data yang belum pernah dilihat sebelumnya.

Selain itu, pada Gambar 3, ditampilkan contoh dari masing-masing jenis jerawat yang ada dalam dataset ini, yang memberikan gambaran lebih jelas tentang variasi jerawat yang diklasifikasikan. Gambar tersebut memperlihatkan variasi data yang digunakan untuk mengklasifikasikan jenis jerawat yang terkadang memiliki perbedaan halus. Meskipun dataset yang digunakan mungkin kurang representatif dalam hal jumlah dan variasi, diharapkan model tetap mampu memberikan klasifikasi dengan tingkat akurasi yang tinggi, sehingga dapat membantu proses identifikasi dan penanganan jerawat dengan lebih efektif dan tepat



Gambar 3. Contoh 3 jenis jerawat

Training model

Proses pelatihan model dimulai dengan memasukkan gambar sebagai input, baik gambar asli maupun yang telah diproses sebelumnya. Langkah awal ini sangat penting karena kualitas gambar yang diolah akan mempengaruhi keseluruhan kinerja model. Setelah gambar diterima, tahap berikutnya adalah pra-pemrosesan, yang melibatkan beberapa langkah kunci seperti mengubah ukuran gambar, normalisasi nilai piksel dan augmentasi data. Pertama, gambar diubah ukurannya menjadi 224x224 piksel agar sesuai dengan kebutuhan ResNet-50. Ukuran ini dipilih karena model ResNet-50 dirancang khusus untuk bekerja dengan gambar berukuran tersebut, sehingga dapat memproses data dengan efisien.

Selain mengubah ukuran, langkah normalisasi juga menjadi bagian penting dalam pra-pemrosesan. Pada tahap ini, nilai piksel gambar disesuaikan ke dalam rentang yang lebih seragam untuk memastikan bahwa model dapat mengenali pola dengan lebih baik. Dengan normalisasi, perbedaan skala antara nilai piksel dapat diminimalkan, sehingga model dapat lebih fokus pada karakteristik penting dari gambar tersebut tanpa terganggu oleh variasi yang tidak relevan[16]. Proses ini juga membantu mempercepat pelatihan dengan memastikan distribusi data yang lebih konsisten, memungkinkan model untuk mencapai hasil yang optimal dengan lebih cepat dan akurat.

Setelah normalisasi yang menjaga nilai piksel tetap seragam, ada langkah-langkah lain dalam pra-pemrosesan yang dirancang untuk memastikan kualitas data sebelum masuk ke tahap pelatihan. Salah satunya adalah augmentasi data, metode augmentasi data yang digunakan dalam penelitian ini mencakup beberapa teknik yang dirancang untuk meningkatkan variasi dalam dataset dan memperkuat kemampuan model dalam mengenali pola dari berbagai kondisi. Metode augmentasi tersebut meliputi *rescale* (menyesuaikan skala nilai piksel), *rotate* (memutar gambar), *zoom* (memperbesar atau memperkecil gambar), *flip* (membalik gambar), *width shift* (menggeser gambar secara horizontal), *fill mode nearest* (mengisi area kosong dengan piksel terdekat), dan *shear* (menggeser bagian gambar secara miring).

Penggunaan metode-metode ini bertujuan untuk memastikan bahwa model tidak hanya dilatih dengan data yang terbatas pada kondisi tertentu, tetapi juga mampu mengenali pola-pola dari gambar baru yang mungkin berbeda dari data pelatihan. Augmentasi ini membantu memperluas variasi dataset, membuat model lebih tangguh dan akurat dalam mengklasifikasikan jenis jerawat di berbagai kondisi pencahayaan, sudut, dan skala. Dengan demikian, model dapat fokus pada fitur-fitur utama yang diperlukan untuk klasifikasi yang lebih andal.

Setelah pra-pemrosesan, citra kemudian diproses melalui serangkaian lapisan deteksi fitur. Lapisan-lapisan ini terdiri dari CL, *pooling*, dan *Rectified Linear Unit* (ReLU). CL menerapkan filter pada citra input untuk mengekstrak fitur-fitur relevan seperti tepi, kurva, atau informasi warna. Bobot pada filter menentukan seberapa penting setiap piksel dalam mendeteksi fitur tersebut. Hasil konvolusi adalah citra baru yang merupakan representasi linier dari citra asli, namun dengan penekanan pada fitur-fitur yang sesuai dengan filter yang dipilih [22]. Selanjutnya, lapisan *pooling* melakukan *downsampling* data untuk mengurangi dimensi dan jumlah parameter, sehingga mengurangi kompleksitas komputasi. Fungsi aktivasi ReLU, yang mengubah semua nilai negatif menjadi nol dan mempertahankan nilai positif, membantu mengatasi masalah hilangnya gradien. Gradien mati adalah fenomena di mana gradien yang ditransmisikan kembali ke lapisan sebelumnya

menjadi sangat kecil, mengakibatkan penurunan kecepatan *konvergensi* dan kualitas pelatihan model yang buruk [23].

Setelah fitur-fitur diekstraksi, mereka diproses melalui lapisan klasifikasi yang terdiri FCL dan fungsi aktivasi *softmax*. FCL menghubungkan semua neuron pada lapisan sebelumnya dengan neuron pada lapisan output, memungkinkan model untuk membuat prediksi berdasarkan fitur-fitur yang telah diekstraksi. Fungsi *softmax* digunakan untuk memetakan angka yang didapat dari perhitungan FCL ke dalam sebuah probabilitas yang menunjukkan kelas dari citra tersebut. Tahap ini penting karena menentukan kategori akhir dari citra yang diinput berdasarkan probabilitas tertinggi yang dihasilkan [24].

Optimasi model dilakukan menggunakan metode Adam (*Adaptive Moment Estimation*), yang dikenal karena kecepatannya dalam mencapai konvergensi dan kemampuannya mengatasi lambatnya konvergensi pada metode *Stochastic Gradient Descent* (SGD). Adam menghitung kecepatan pembelajaran adaptif untuk setiap parameter [25]. Optimasi ini dipilih karena dapat menangani perubahan parameter secara efisien, bahkan dalam kondisi di mana data pelatihan bervariasi akibat augmentasi yang diterapkan. Hal ini penting untuk memastikan bahwa model dapat belajar secara efektif dari dataset yang telah diperluas variasinya, sehingga mampu mengidentifikasi jenis jerawat dengan lebih akurat.

Selain itu, teknik dropout digunakan untuk mengurangi *overfitting* pada *neural networks*. Dropout bekerja dengan memilih secara acak dan membuang neuron-neuron yang tidak dipakai selama proses training. Teknik ini tidak hanya membantu dalam mencegah adaptasi kompleks yang bisa menyebabkan *overfitting*, tetapi juga mempercepat proses learning. Dropout memungkinkan model untuk lebih robust dengan memastikan bahwa tidak ada neuron yang terlalu bergantung pada neuron lain [26].

Pada tahap akhir, model menghasilkan vektor probabilitas yang menunjukkan probabilitas setiap kelas untuk citra yang diinput. Kelas dengan probabilitas tertinggi dianggap sebagai kategori yang paling mungkin untuk gambar tersebut. Output ini memberikan hasil akhir dari proses klasifikasi, memungkinkan pengguna untuk memahami prediksi model berdasarkan input citra yang diberikan.

Testing

Setelah model CNN ResNet-50 dilatih, tahap selanjutnya adalah mengevaluasi kinerjanya dengan dataset testing. Proses evaluasi ini sangat penting karena menunjukkan seberapa baik model dapat menggeneralisasi pola yang telah dipelajari dari data pelatihan ke data baru yang belum pernah dilihat. Dataset testing berfungsi sebagai indikator untuk mengukur performa model di situasi dunia nyata, di mana model harus bisa mengenali jenis jerawat dengan akurasi tinggi meskipun gambar yang dihadapi memiliki variasi yang tidak terduga.

Proses evaluasi dimulai dengan menjalankan dataset testing melalui model yang telah dilatih. Model kemudian memproses setiap gambar dan menghasilkan prediksi untuk kelas jerawat yang sesuai. Prediksi ini dibandingkan dengan label asli yang telah ditentukan sebelumnya. Dari perbandingan ini, berbagai metrik evaluasi dihitung, seperti *loss* dan akurasi. *Loss* memberikan gambaran tentang kesalahan prediksi model, sedangkan akurasi menunjukkan persentase prediksi yang benar dibandingkan dengan total gambar yang diuji.

Penggunaan metrik ini tidak hanya memberikan informasi tentang seberapa baik model bekerja secara keseluruhan, tetapi juga membantu mengidentifikasi apakah ada aspek tertentu yang perlu ditingkatkan, seperti kemampuan model dalam menangani jenis gambar yang lebih kompleks atau kondisi pencahayaan yang berbeda. Selain itu, evaluasi ini penting untuk mendeteksi potensi *overfitting*, di mana model bekerja sangat baik pada data pelatihan tetapi kurang efektif pada data baru.

Selain itu, *confusion matrix* digunakan sebagai alat evaluasi untuk mengukur kinerja klasifikasi model dengan lebih rinci. *Confusion matrix* adalah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan lebih rinci, menggambarkan bagaimana prediksi model tersebar di setiap kelas [10]. Ada empat istilah yang merepresentasikan hasil klasifikasi saat mengukur performa menggunakan *confusion matrix*, yaitu *Total True Positive* (TTP), *Total True Negative* (TTN), *Total False Positive* (TFP), dan *Total False Negative* (TFN). *Confusion matrix* memungkinkan kita untuk menghitung metrik tambahan seperti *accuracy*, *precision*, *recall*, dan *F1-score*, yang memberikan wawasan lebih mendalam tentang bagaimana model menangani kelas yang jarang serta kemampuannya dalam memprediksi kelas dengan akurat.

Akurasi menggambarkan sejauh mana sistem dapat mengklasifikasikan data dengan benar. Nilai akurasi dihitung sebagai rasio antara jumlah data yang diklasifikasikan dengan benar terhadap total jumlah data yang ada. Akurasi ini dapat diperoleh melalui Persamaan (1) berikut.

$$accuracy = \frac{TTP_{all} + TTN_{all}}{Jumlah\ Total\ Pengujian} \quad (1)$$

Presisi merupakan rasio antara jumlah data dalam suatu kategori yang diklasifikasikan dengan benar terhadap total data yang terklasifikasi dalam kategori tersebut. Nilai presisi dapat dihitung menggunakan Persamaan (2) berikut.

$$presisi = \frac{TTP_{all}}{TTP_{all} + TFP_i} \quad (2)$$

Recall digunakan untuk mengetahui rasio antara jumlah data dalam suatu kategori yang diklasifikasikan dengan benar oleh sistem dengan total data dalam kategori tersebut, baik yang terklasifikasi dengan benar maupun salah. Nilai recall dapat dihitung menggunakan Persamaan (3) berikut.

$$recall = \frac{TTP_{all}}{TTP_{all} + TFN_i} \quad (3)$$

F1-Score adalah metrik yang menggabungkan presisi dan recall. Nilai F1-Score dapat dihitung menggunakan Persamaan (4) berikut.

$$f1 - score = 2x \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$

TTP_{all} merujuk pada total data yang menunjukkan situasi ketika data positif berhasil diprediksi dengan benar sebagai positif. TFN_{all} menggambarkan kondisi di mana data negatif diprediksi dengan benar sebagai negatif. TFP_i mengindikasikan situasi ketika data negatif salah diprediksi sebagai positif, sedangkan TFN_i menggambarkan kondisi ketika data positif salah diprediksi sebagai negatif [27].

Evaluasi

Setelah model melewati tahap pelatihan dan *testing*, langkah berikutnya adalah melakukan analisis mendalam terhadap hasil testing untuk menilai performa model secara menyeluruh. Proses analisis ini dimulai dengan memeriksa metrik kinerja utama, seperti akurasi, untuk mendapatkan gambaran umum tentang seberapa sering model berhasil membuat prediksi yang benar pada data yang belum pernah dilihat sebelumnya. Akurasi memberikan indikasi dasar tentang kinerja model, tetapi tidak selalu mengungkapkan semua detail terkait efektivitasnya.

Jika hasil menunjukkan akurasi yang memuaskan dan distribusi *error* dalam *confusion matrix* minim, maka model mungkin siap untuk diterapkan. Namun, jika *loss* cukup tinggi atau akurasi tidak memadai, ini mengindikasikan bahwa model perlu diperbaiki. Misalnya, jika FP atau FN terlalu tinggi pada kelas tertentu, ini bisa menjadi sinyal bahwa model kesulitan dalam membedakan kelas tersebut. Pada titik ini, langkah-langkah analisis lebih mendalam diperlukan, seperti memeriksa apakah model terlalu kompleks (yang menyebabkan *overfitting*) atau terlalu sederhana (mengakibatkan *underfitting*).

Analisis kesalahan juga sangat penting dalam proses evaluasi ini. Dengan memeriksa kasus di mana model membuat kesalahan, kita dapat memahami pola kesalahan dan area yang memerlukan perbaikan. Misalnya, jika model sering salah mengidentifikasi jenis jerawat tertentu, ini mungkin menandakan bahwa data pelatihan kurang representatif atau fitur visual yang digunakan oleh model kurang sesuai. *Confusion matrix* akan menunjukkan kelas mana yang sering salah diklasifikasikan dan memberikan wawasan lebih detail mengenai performa model.

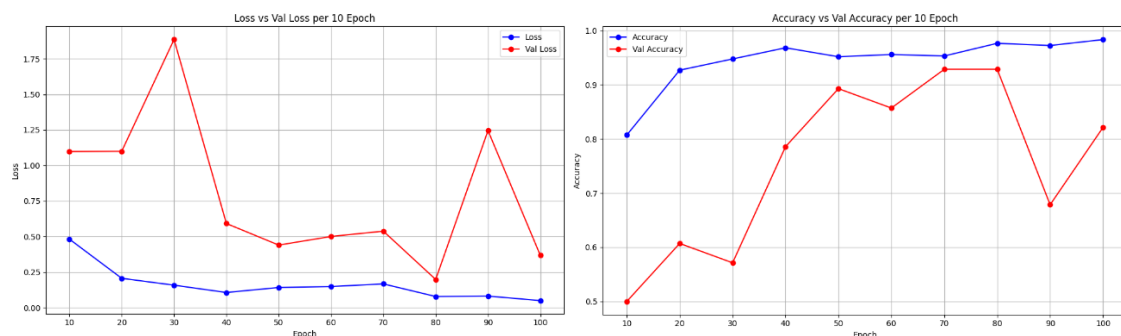
Melalui analisis yang mendalam terhadap hasil testing, termasuk penggunaan *confusion matrix*, dapat diidentifikasi area yang perlu diperbaiki untuk meningkatkan kinerja keseluruhan model. Proses ini memastikan bahwa model tidak hanya siap untuk digunakan, tetapi juga mampu memberikan hasil yang akurat dan andal di dunia nyata. Evaluasi berkelanjutan serta penyesuaian yang diperlukan akan memastikan model bekerja secara optimal dan mampu menangani berbagai kondisi yang mungkin ditemui dalam aplikasi praktis.

III. HASIL DAN PEMBAHASAN

Sebagaimana telah dijelaskan dalam metode penelitian, dataset yang digunakan berjumlah 754 gambar yang terbagi menjadi tiga kelas utama, dari jumlah tersebut dataset terbagi menjadi 726 data untuk pelatihan dan 28 data untuk pengujian. Langkah awal dalam pelatihan adalah melakukan preprocessing gambar dengan mengubah parameter nilai piksel dan melakukan augmentasi data untuk meningkatkan variasi dalam dataset. Kemudian, parameter-parameter model diatur: model dilatih selama 100 *epoch*, menggunakan dropout 20%

untuk mengurangi risiko *overfitting*, dan dioptimalkan dengan optimizer Adam untuk mengurangi fungsi kerugian. Sebuah *checkpoint* model juga diimplementasikan untuk menyimpan hasil terbaik selama pelatihan. Setelah penyesuaian parameter selesai, proses berikutnya adalah melatih dan menguji model menggunakan dataset yang telah disiapkan.

Pada Gambar 4, grafik menunjukkan perbandingan antara *loss* dengan *validation loss* (*val_loss*) serta *accuracy* dengan *validation accuracy* (*val_accuracy*) selama 100 epoch, dengan data diambil setiap 10 epoch. *Loss* mengukur seberapa baik model memprediksi, di mana nilai yang lebih rendah menunjukkan prediksi yang lebih baik. Sementara itu, *val_loss* mengukur kinerja model pada data yang tidak digunakan selama pelatihan. Kenaikan *val_loss* dibanding *loss* dapat mengindikasikan *overfitting*. *Accuracy* menggambarkan persentase prediksi benar pada data pelatihan, sedangkan *val_accuracy* mengukur hal serupa pada data validasi. Tren peningkatan pada *accuracy* dan *val_accuracy* menunjukkan model terus belajar, meskipun ada fluktuasi yang disebabkan oleh kompleksitas data.



Gambar 4. Hasil Pelatihan Model CNN Resnet-50

Grafik ini menunjukkan penurunan yang konsisten pada *loss*, menandakan bahwa model semakin baik dalam memprediksi data pelatihan. Namun, *val_loss* mengalami beberapa fluktuasi, dengan sesekali peningkatan. *Accuracy* pada data pelatihan terus meningkat, sedangkan *val_accuracy* menunjukkan tren peningkatan yang lebih stabil meskipun ada beberapa penurunan di titik-titik tertentu. Grafik ini mengilustrasikan proses pembelajaran model dari data pelatihan dan kemampuannya untuk menangani data baru. Interpretasi ini menekankan pentingnya memantau kinerja model pada data validasi agar tetap efektif dalam mengolah data yang belum pernah dilihat sebelumnya.

Selain itu, disajikan juga Tabel 1 yang memuat hasil pelatihan model secara rinci dari percobaan yang sama, termasuk tambahan informasi mengenai waktu per langkah atau waktu yang dihabiskan pada setiap epoch. Tabel ini menampilkan nilai-nilai mengenai performa model selama pelatihan.

TABEL 1
HASIL PELATIHAN MODEL CNN RESNET-50

Epoch	Waktu per Langkah	Loss	Accuracy	Val loss	Val Accuracy
10	13s 279ms/step	0.4813	0.8072	1.0978	0.5000
20	12s 250ms/step	0.2056	0.9270	1.0995	0.6071
30	12s 262ms/step	0.1575	0.9477	1.8858	0.5714
40	13s 277ms/step	0.1059	0.9683	0.5912	0.7857
50	13s 282ms/step	0.1411	0.9518	0.4395	0.8929
60	13s 282ms/step	0.1478	0.9559	0.5000	0.8571
70	14s 300ms/step	0.1663	0.9532	0.5374	0.9286
80	13s 284ms/step	0.0778	0.9766	0.1976	0.9286
90	13s 276ms/step	0.0806	0.9725	1.2446	0.6786
100	12s 260ms/step	0.0489	0.9835	0.3670	0.8214

Dapat dilihat dari Gambar 4 maupun Tabel 1 hasil pelatihan model dengan berbagai jumlah epoch menunjukkan perbedaan antara *loss* dan *val_loss* serta antara *accuracy* dan *val_accuracy* pada setiap epoch-nya. Secara keseluruhan, hasil optimal terlihat pada akhir pelatihan di *epoch* 80 dan 100. Pada *epoch* 80, nilai *loss* pada data pelatihan mencapai 0.0778 dengan akurasi 0.9766, sementara *val_loss* stabil pada 0.1976 dengan

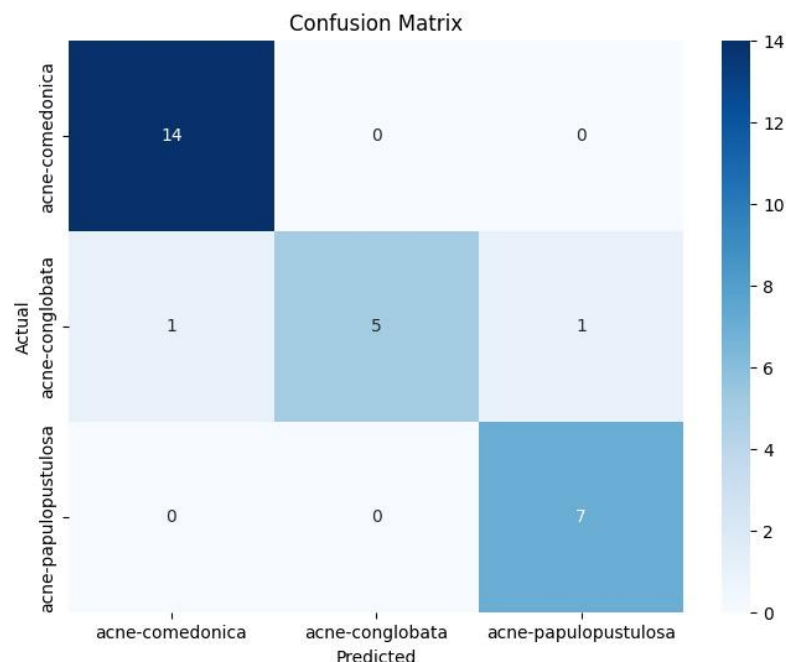
val_accuracy 0.9286. Peningkatan ini menunjukkan bahwa model mampu mengurangi kesalahan prediksi dengan baik pada kedua set data, baik pelatihan maupun validasi. Di *epoch* 100, meskipun terjadi penurunan kecil dalam *loss* menjadi 0.0489 dan kenaikan dalam akurasi menjadi 0.9835 pada data pelatihan, *val_loss* tetap rendah pada 0.3670 dengan *val_accuracy* 0.8214, menunjukkan bahwa model masih mampu mempertahankan kualitasnya dalam menggeneralisasi pada data yang belum pernah dilihat sebelumnya.

Meskipun model menunjukkan peningkatan akurasi dan penurunan *loss* seiring bertambahnya epoch, perbedaan signifikan antara *loss* dan *val_loss* pada epoch ke-90 menandakan potensi *overfitting*. Pada epoch ini, *loss* pada data pelatihan sangat rendah, mencapai 0.0806, yang menunjukkan bahwa model sangat baik dalam memprediksi data pelatihan. Namun, *val_loss* meningkat drastis menjadi 1.2446, menunjukkan bahwa model tidak dapat menerapkan pengetahuannya dengan sama baiknya pada data validasi.

Perbedaan ini terjadi karena model mulai terlalu spesifik pada pola-pola yang ada di data pelatihan, bahkan pada noise atau detail yang tidak relevan, sehingga model menghafal data tersebut alih-alih memahami pola yang lebih umum. Ketika dihadapkan pada data validasi yang belum pernah dilihat sebelumnya, model gagal menggeneralisasi pengetahuannya, yang ditandai dengan peningkatan *val_loss*. *Overfitting* ini terjadi karena model terlalu terlatih pada data pelatihan, yang menyebabkan performa pada data yang lebih bervariasi (data validasi) menurun. Oleh karena itu, perbedaan signifikan ini menjadi indikator kuat bahwa model mungkin sudah mulai mengalami *overfitting*.

Penggunaan *checkpoint* dalam pelatihan model sangat penting, karena membantu dalam menyimpan model terbaik berdasarkan performa validasi. Hal ini memungkinkan kita untuk kembali ke model dengan performa optimal sebelum terjadinya *overfitting*. Selain mengukur *accuracy*, *val_accuracy*, *loss*, dan *val_loss*, pengukuran *confusion matrix* juga dilakukan untuk mengevaluasi kinerja model secara lebih mendalam, mengingat bahwa model pada *epoch* ke-80 memiliki akurasi validasi tertinggi dan *val_loss* terendah. Dengan demikian, analisis pada *epoch* ke-80 menjadi relevan untuk mendapatkan wawasan yang lebih baik tentang bagaimana model mengklasifikasikan data.

Pada gambar 5 menampilkan *confusion matrix* dari model CNN ResNet-50 *epoch* ke-80 yang digunakan untuk klasifikasi jenis jerawat. Confusion matrix ini menggambarkan kinerja model dalam mengklasifikasikan tiga kelas utama: *acne-comedonica*, *acne-conglobata*, dan *acne-papulopustulosa*. Pada sumbu horizontal terdapat prediksi kelas, sedangkan pada sumbu vertikal menunjukkan kelas aktual. Terdapat 14 data tes untuk *acne-comedonica*, 7 untuk *acne-conglobata*, dan 7 untuk *acne-papulopustulosa*.



Gambar 5. Confusion Matrix Model Latih Epoch ke-80

Untuk kelas *acne-comedonica*, model berhasil mengklasifikasikan semua 14 gambar dengan benar, terlihat dari nilai 14 di sel yang sesuai. Kelas *acne-conglobata* menunjukkan beberapa kesalahan, di mana 5

dari 7 gambar yang benar diklasifikasikan, sementara 1 gambar diklasifikasikan sebagai *acne-comedonica* dan 1 gambar sebagai *acne-papulopustulosa*. Untuk kelas *acne-papulopustulosa*, model juga menunjukkan hasil yang baik dengan 7 dari 7 gambar di klasifikasikan dengan benar. Secara keseluruhan, gambar ini mengilustrasikan bahwa model memiliki kinerja yang baik, terutama dalam mengklasifikasikan kelas *acne-comedonica* dan *acne-papulopustulosa*, tetapi terdapat beberapa tantangan dalam pengenalan kelas *acne-conglobata*, yang mungkin disebabkan oleh kesamaan visual antara kelas-kelas tersebut.

Disajikan juga gambar 6 yang menampilkan hasil evaluasi model CNN ResNet-50 pada *checkpoint epoch* ke-80 dengan menggunakan metrik *precision*, *recall*, dan *F1-score* untuk tiga kelas utama. Tabel ini juga mencantumkan nilai support untuk setiap kelas, yang merepresentasikan jumlah gambar yang diuji per kelas. Terdapat juga akurasi keseluruhan model serta nilai rata-rata *precision*, *recall*, dan *F1-score* dalam dua kategori, yaitu *macro average* dan *weighted average*.

	precision	recall	f1-score	support
acne-comedonica	0.93	1.00	0.97	14
acne-conglobata	1.00	0.71	0.83	7
acne-papulopustulosa	0.88	1.00	0.93	7
accuracy			0.93	28
macro avg	0.94	0.90	0.91	28
weighted avg	0.94	0.93	0.92	28

Gambar 6. Hasil Evaluasi Confusion Matrix Model Latih Epoch ke-80

Hasil evaluasi menunjukkan bahwa model CNN ResNet-50 memiliki performa yang sangat baik dalam klasifikasi jenis jerawat, dengan akurasi keseluruhan mencapai 93%. Kelas *acne-comedonica* menunjukkan hasil optimal dengan *precision* 0.93 dan *recall* 1.00, menandakan bahwa model sangat akurat dalam mengenali gambar dari kelas ini. Meskipun *precision* untuk kelas *acne-conglobata* mencapai nilai sempurna (1.00), penurunan *recall* menjadi 0.71 menunjukkan tantangan dalam mengenali sebagian gambar dari kelas tersebut, yang mengakibatkan *F1-score* yang lebih rendah (0.83). Kelas *acne-papulopustulosa* menunjukkan hasil yang baik dengan *precision* 0.88 dan *recall* 1.00, menghasilkan *F1-score* 0.93. Dengan *macro average F1-score* sebesar 0.91 dan *weighted average F1-score* 0.92, hasil ini menunjukkan bahwa meskipun terdapat beberapa tantangan, secara keseluruhan, model mampu memberikan klasifikasi yang akurat dan konsisten pada berbagai jenis jerawat.

IV. KESIMPULAN

Dalam penelitian ini, dapat disimpulkan bahwa CNN dengan arsitektur ResNet-50 mampu melakukan klasifikasi jenis jerawat pada gambar kulit wajah dengan tingkat akurasi yang tinggi. Hasil pelatihan menunjukkan akurasi mencapai 98.35% dengan nilai *loss* sebesar 0.0489. Model juga berhasil mencapai tingkat validasi akurasi tertinggi sebesar 92.86% dengan *loss* validasi sebesar 0.1976. Dari analisis confusion matrix pada epoch ke-80, model menunjukkan akurasi keseluruhan sebesar 93%. Model berhasil mengklasifikasikan kelas *acne-comedonica* dengan *precision* 0.93, *recall* 1.00, dan *F1-score* 0.97, serta *acne-papulopustulosa* dengan *precision* 0.88, *recall* 1.00, dan *F1-score* 0.93. Namun, untuk kelas *acne-conglobata*, meskipun *precision*-nya sempurna sebesar 1.00, *recall* yang lebih rendah sebesar 0.71 menyebabkan *F1-score* menurun menjadi 0.83. Hasil evaluasi metrik *precision*, *recall*, dan *F1-score* menunjukkan bahwa meskipun model memiliki performa yang sangat baik secara keseluruhan, ada ruang untuk perbaikan, terutama pada kelas *acne-conglobata*, yang memiliki *precision* sempurna tetapi *recall* yang lebih rendah. Indikasi *overfitting* juga teramati pada beberapa epoch, yang berdampak pada stabilitas akurasi validasi. Oleh karena itu, penggunaan teknik *hyperparameter tuning*, seperti *grid search* atau *random search*, sangat disarankan untuk menemukan kombinasi *parameter* yang optimal, seperti *learning rate*, ukuran *batch*, dan tingkat *dropout*. Dengan demikian, hal ini diharapkan dapat meminimalkan *overfitting* dan meningkatkan kinerja model secara keseluruhan, serta memastikan kemampuan model dalam menggeneralisasi data baru dengan lebih efektif.

DAFTAR PUSTAKA

- [1] N. Sifatullah and Zulkarnain, "Jerawat (Acne vulgaris): Review Penyakit Infeksi Pada Kulit," *Prosiding Biologi Achieving the Sustainable Development Goals*, no. November, pp. 19–23, 2021, [Online]. Available: <http://journal.uin-alauddin.ac.id/index.php/psb>
- [2] T. IMASARI and F. Emasari, "Deteksi Bakteri Staphylococcus Sp. Penyebab Jerawat Dengan Tingkat Pengetahuan Perawatan Wajah Pada Siswa Kelas Xi Di Smk Negeri 1 Pagerwojo," *Jurnal Sintesis: Penelitian Sains, Terapan dan Analisisnya*, vol. 2, no. 2, pp. 58–65, 2022, doi: 10.56399/jst.v2i2.20.
- [3] D. Anggraeni, M. Kaniawati, and G. Jafar, "Pendekatan Nanoteknologi Untuk Penghantaran Bahan Aktif Farmasi Dalam Terapi Acne Vulgaris," *Majalah Farmasetika*, vol. 8, no. 4, p. 283, 2023, doi: 10.24198/mfarmasetika.v8i4.45498.
- [4] S. A. Chintya, S. Khomastin, and L. Farida, "Pengaruh Tingkat Kecemasan Sosial, Depresi Dan Kualitas Hidup Terhadap Acne Vulgaris," *ULIL ALBAB: Jurnal Ilmiah Multidisiplin*, vol. 1, no. 6, pp. 1457–1463, 2022.
- [5] E. Topang, A. Soekanto, M. W. Sugeng, and R. D. Wulandari, "Hubungan Stres Dengan Kejadian Akne Vulgaris Pada Remaja (Studi Literatur)," *Calvaria Medical Journal*, vol. 2, no. 1, pp. 36–41, 2024, doi: 10.30742/cmj.v2i1.30.
- [6] E. Suryokta, W. Taruklimbong, and H. Sihotang, "Peluang dan Tantangan Penggunaan AI (Artificial Intelligence) dalam Pembelajaran Kimia," *Jurnal Pendidikan Tambusai*, vol. 7, no. 3, pp. 26745–26757, 2023.
- [7] B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, "Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecenderungan Warna dengan Metode Deep Learning Convolutional Neural Network," *INOVTEK Polbeng - Seri Informatika*, vol. 6, no. 2, p. 259, 2021, doi: 10.35314/isi.v6i2.2104.
- [8] D. Aprillia, T. Rohana, T. Al Mudzakir, and ..., "Deteksi Nominal Mata Uang Rupiah Menggunakan Metode Convolutional Neural Network dan Feedforward Neural Network," *KLIK: Kajian Ilmiah ...*, vol. 4, no. 4, pp. 2068–2077, 2024, doi: 10.30865/klik.v4i4.1711.
- [9] N. Mamuriyah and J. Sumantri, "Penerapan Metode Convolution Neural Network (CNN) Pada Aplikasi Automatic Lip Reading," *Journal of Informatics and Telecommunication Engineering*, vol. 6, no. 1, pp. 276–287, 2022, doi: 10.31289/jite.v6i1.7523.
- [10] F. Paraijun, R. N. Aziza, and D. Kuswardani, "Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Kesegaran Buah Berdasarkan Citra Buah," *Kilat*, vol. 11, no. 1, pp. 1–9, 2022, doi: 10.33322/kilat.v10i2.1458.
- [11] I. Suhardin, A. Patombongi, and A. M. Islah, "Mengidentifikasi Jenis Tanaman Berdasarkan Citra Daun Menggunakan Algoritma Convolutional Neural Network," *Simtek : jurnal sistem informasi dan teknik komputer*, vol. 6, no. 2, pp. 100–108, 2021, doi: 10.51876/simtek.v6i2.101.
- [12] M. O. Nugraha, R. Purnamasari, and S. Aulia, "Klasifikasi Penyakit Berdasarkan Warna Kuku Menggunakan Pengolahan Sinyal Digital (Classification Of Diseases Based On Nail Color Using Digital Signal Processing)," *e-Proceeding of Engineering*, vol. 8, no. 6, p. 3226, 2022.
- [13] P. Mbaba, K. Anwar, S. Rahayu, A. Eka Kartawati, P. Studi Informatika, and F. Matematika dan Ilmu Pengetahuan Alam Universitas Udayana Bali, "Penerapan Metode Convolution Neural Networks Untuk Mengidentifikasi Wajah Kelelahan," *Jnatia*, vol. 1, no. 1, pp. 107–114, 2022, [Online]. Available: <https://www.kaggle.com/datasets/davidvazquez/cic/yawn-dataset>
- [14] K. Azmi, S. Defit, and S. Sumijan, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *Jurnal Unitek*, vol. 16, no. 1, pp. 28–40, 2023, doi: 10.52072/unitek.v16i1.504.
- [15] Micheal and E. Hartati, "Klasifikasi spesies kupu kupu menggunakan metode convolutional neural network," *MDP Student Conference2022*, vol. 1, no. 1, pp. 569–577, 2022, [Online]. Available: <https://jurnal.mdp.ac.id/index.php/msc/article/view/1928/541>

- [16] E. H. Rachmawanto, D. Hermanto, Z. Pratama, and C. A. Sari, "Performa Convolutional Neural Network Dalam Deep Layers Resnet-50 Untuk Klasifikasi Mri Tumor Otak," *Semnas Ristek (Seminar Nasional Riset dan Inovasi Teknologi)*, vol. 8, no. 01, pp. 6–12, 2024, doi: 10.30998/semnasristek.v8i01.7125.
- [17] S. P. Ristiawanto, B. Irawan, and C. Setianingsih, "Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network Menggunakan Arsitektur Residual Network-50," *e-Proceeding of Engineering*, vol. 8, no. 5, pp. 6442–6454, 2021.
- [18] S. Kirana Wulandari and Jasmir, "Penggunaan resnet-50 untuk deteksi penyakit ikan air tawar di akuakultur studi kasus pada akuakultur asia selatan," *SENABISTEKES*, vol. 1, pp. 17–24, 2024, [Online]. Available: <https://ejournal.ummuba.ac.id/index.php/SENABISTEKES/article/view/2205/1113>
- [19] I. Hasan, Suprayogi, and H. B. D, "Klasifikasi Jenis Jerawat Menggunakan Convolutional Neural Networks," *e-Proceeding of Engineering*, vol. 8, no. 1, pp. 358–372, 2021.
- [20] F. Sudana Putra, Kusriani, and M. P. Kurniawan, "Deteksi Otomatis Jerawat Wajah Menggunakan Metode Convolutional Neural Network (CNN)," *Journal of Information Technology*, vol. 1, no. 2, pp. 30–34, 2021, doi: 10.46229/jifotech.v1i2.308.
- [21] N. IBRAHIM *et al.*, "Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 1, p. 162, 2022, doi: 10.26760/elkomika.v10i1.162.
- [22] T. Hidayat, N. Khasanah, D. U. E. Saputri, U. Khultsum, and R. L. Pratiwi, "Klasifikasi Gambar Palmprint Berbasis Multi-Kelas Menggunakan Convolutional Neural Network," *Jurnal Sistem Informasi*, vol. 11, no. 1, pp. 01–06, 2022, doi: 10.51998/jsi.v11i1.474.
- [23] M Mesran, Sitti Rachmawati Yahya, Fifto Nugroho, and Agus Perdana Windarto, "Investigating the Impact of ReLU and Sigmoid Activation Functions on Animal Classification Using CNN Models," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 1, pp. 111–118, 2024, doi: 10.29207/resti.v8i1.5367.
- [24] R. Magdalena, S. Saidah, N. K. C. Pratiwi, and A. T. Putra, "Klasifikasi Tutupan Lahan Melalui Citra Satelit SPOT-6 dengan Metode Convolutional Neural Network (CNN)," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 7, no. 3, p. 335, 2021, doi: 10.26418/jp.v7i3.48195.
- [25] N. Faridah and B. Sugiantoro, "Analisis Optimasi Pada Algoritma Long Short Term Memory Untuk Memprediksi Harga Saham," *Jurnal Media Informatika Budidarma*, vol. 7, no. 1, pp. 575–582, 2023, doi: 10.30865/mib.v7i1.5421.
- [26] I. G. T. Suryawan and I. P. A. E. Darma Udayana, "Optimasi Convolutional Neural Network Untuk Deteksi Covid-19 pada X-ray Thorax Berbasis Dropout," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 3, p. 551, 2022, doi: 10.25126/jtiik.2022935143.
- [27] R. AGUSTINA, R. MAGDALENA, and N. K. C. PRATIWI, "Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 2, p. 446, 2022, doi: 10.26760/elkomika.v10i2.446.