

# KLASIFIKASI JENIS JERAWAT MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORKS

Ibrahim Hasan, Suprayogi, Hertiana Bethaningtyas D.

Prodi S1 Teknik Fisika, Fakultas Teknik Elektro, Universitas Telkom

Prodi S1 Teknik Fisika, Fakultas Teknik Elektro, Universitas Telkom

Prodi S1 Teknik Fisika, Fakultas Teknik Elektro, Universitas Telkom

ibrahimhasan@telkomuniversity.ac.id, suprayogi@telkomuniversity.ac.id, hertiana@telkomuniversity.ac.id

## Abstrak

Topik penelitian ini adalah tentang klasifikasi jerawat dengan menggunakan metode Convolutional Neural Networks (CNN). Salah satu alasan utama mengapa judul ini diangkat sebagai topik penelitian adalah di era *digital artificial intelegent* ini sangat dibutuhkan kapasitas untuk mengklasifikasi jerawat dengan menggunakan metode *machine learning* terutama bagi pihak yang memiliki keperluan untuk mengetahui jenis jerawat. Hal ini penting karena tidak semua orang memiliki kemampuan dalam mengklasifikasi jenis jerawat sehingga sangat diperlukan keahlian dalam mengklasifikasi jenis jerawat. Mengingat para ahli di bidang pengklasifikasian jerawat ini sangatlah langka. Tugas Akhir ini menggunakan theoretical framework *deep learning*. Salah satu metode yang digunakan adalah *Convolutional Neural Network* (CNN). Hal yang membedakan *CNN* dengan metode *neural network* lainnya adalah jumlah *hidden layer* yang banyak pada proses klasifikasi. Dalam penelitian ini digunakan 1200 *dataset* jerawat dengan jumlah data *train* dan data *test* masing-masing sebanyak 1070 dan 130 citra. *Preprocessing* data, klasifikasi *CNN*, dan pembuatan model dapat dilewati dengan baik. Hasil yang diperoleh adalah model dapat mengenali dan mengklasifikasikan data citra uji dengan akurasi sebesar 91.6%.

Kata kunci: *Jerawat, Convolutional Neural Networks (CNN), Stochastic Gradient Descent*

## Abstract

The topic of this research is about the classification of acne using the Convolutional Neural Networks (CNN) method. One of the main reasons why this title was raised as a research topic is that in this digital era of artificial intelligence, the capacity to classify acne using machine learning methods is needed, especially for those who have a need for acne types. This is important because not everyone has the ability to classify types of acne so that expertise is needed in classifying types of acne. Providing experts in the field of acne classification is extremely rare. In this case, the theoretical framework used in learning. One of the methods used is Convolutional Neural Network (CNN). What distinguishes CNN from other methods neural networks is the large number of hidden layers in the classification process. In this study, 1200 acne datasets were used with 1070 and 130 images of train data and test data, respectively. Data preprocessing, CNN classification, and modeling can be well done. The results obtained are the model can and classify image data with an accuracy of 91.6%.

Keywords: *Acne, Convolutional Neural Network (CNN), Stochastic Gradient Descent*

## 1. Pendahuluan

Jerawat adalah penyakit kulit akibat abnormalitas produksi sebum pada kelenjar sebacea yang muncul pada kelenjar minyak pada kulit terlalu aktif. Proses kronik kelenjar kelenjar poliblastea yang ditandai dengan adanya komedo, papul, pustule dan nodul. Jerawat merupakan salah satu penyakit kulit yang paling sering muncul di wajah yang ditandai dengan munculnya bintik-bintik. Selain di wajah, jerawat dapat juga muncul di leher, punggung dan dada. Jerawat bisa dialami oleh siapa saja, baik pria maupun wanita, tetapi kebanyakan dialami pada usia remaja.

Bahkan jerawat diperkirakan dialami 9,4% dari populasi global sehingga menjadikannya penyakit umum kedelapan terbanyak di dunia [1]. Tantangan dalam pembuatan aplikasi pendeteksi jerawat adalah kecilnya jerawat sehingga dibutuhkan algoritma yang tepat untuk pendeteksian melalui ekstraksi fitur dan klasifikasi. Salah satu parameter algoritma yang tepat dapat mengatasi overfitting, kecepatan konvergensi yang tepat dan dapat belajar dari data training dengan tepat.

Penelitian tentang penyakit kulit khususnya jerawat sudah ada dilakukan sebelumnya. Beberapa diantaranya adalah menggunakan ekstraksi fitur tekstur berbasis *histogram* dan *Naive Bayes* untuk mengklasifikasikan jenis penyakit kulit jerawat, herpes atau eczema [2]. Ada juga penelitian lain yaitu sistem deteksi bintik/noda dan jerawat pada wajah menggunakan *Gray Level Co-Occurrence Matrix* (GLCM) untuk ekstraksi fitur dan *Support Vector Machine* (SVM) untuk klasifikasi [3]. Penelitian lain yaitu tentang pendeteksian otomatis jerawat menggunakan model model *Markov Random Field* (MRF) yang menghubungkan deskriptor kromofor [4]. Selain itu ada juga penelitian terhadap lesi jerawat, bekas luka dan kulit normal dengan segmentasi citra menggunakan metode *Otsu Thresholding* [5]. Selanjutnya ada penelitian yang lain yaitu identifikasi jenis jerawat berdasarkan kombinasi segmentasi warna dan analisis tekstur dengan *Binary Large Object* (BLOB) menggunakan klasifikasi *Learning Vector Quantization* [6].

Penelitian-penelitian terdahulu belum banyak yang melakukan klasifikasi jenis jerawat, hanya sebatas deteksi ada tidaknya jerawat dan juga identifikasi jerawat dengan penyakit kulit lainnya, bintik/noda, bekas luka ataupun kulit normal. Ada penelitian terhadap klasifikasi jerawat yang sudah pernah dilakukan, tetapi hasil akurasi masih belum maksimal sehingga penulis merasa perlunya dilakukan penelitian lain terhadap klasifikasi jenis jerawat dengan metode lain untuk mendapat hasil akurasi yang lebih maksimal. Oleh karena itu, dalam penelitian ini penulis mengajukan metode klasifikasi jerawat dengan menggunakan *Convolutional Neural Network* (CNN). Oleh karena itu, maka penulis melakukan penelitian dengan judul “Klasifikasi Jerawat dengan Menggunakan Convolutional Neural Network (CNN)”.

Sejalan dengan permasalahan tersebut maka rumusan penelitian ini adalah:

- a. Arsitektur Convolutional Neural Network seperti apa yang cocok untuk klasifikasi jerawat?
- b. Bagaimana pemilihan pretrained model transfer learning?
- c. Bagaimana implementasi VGGnet pada klasifikasi jerawat?
- d. Bagaimana pemilihan metoda optimasi untuk Gradient Descent yang optimal?
- e. Bagaimana implementasi Stochastic Gradient Descent untuk optimasi Convolutional Neural Networks?

Adapun tujuan yang ingin dicapai dari pembuatan tugas akhir ini adalah:

- a. Merancang desain arsitektur jaringan untuk mengklasifikasi jenis-jenis jerawat dengan menggunakan Convolutional Neural Network (CNN).
- b. Mengimplementasikan struktur jaringan klasifikasi jerawat dengan menggunakan Convolutional Neural Networks (CNN)
- c. Mengukur tingkat akurasi foto klasifikasi jerawat dengan menggunakan Convolutional Neural Networks (CNN)

Metodologi penelitian pada penelitian ini menggunakan berepa langkah yaitu studi literature tentang Convolutional Neural Networks (CNN), analisis permasalahan, perancangan arsitektur CNN, implementasi dan pengujian serta dokumentasi dan pelaporan.

## 2. Dasar Teori, Material dan Metodologi/Perancangan

### 2.1. Convolutional Neural Networks (CNN)

*Convolutional Neural Network* atau dikenal juga dengan sebutan *ConvNets* adalah sebuah metode untuk memproses data dalam bentuk beberapa array, contohnya yaitu gambar berwarna yang terdiri dari tiga array 2D yang mengandung intensitas piksel dalam tiga jenis warna. *Convolutional Neural Networks* (ConvNets) merupakan penerapan dari *Artificial Neural Networks* (ANN) yang lebih istimewa dan saat ini diklaim sebagai model terbaik untuk memecahkan masalah pengenalan objek. Secara teknis, *convolutional network* memiliki arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan dan keluaran dari masing-masing tahap adalah beberapa array yang disebut *feature map* atau peta fitur. Contohnya untuk citra *greyscale*, *input* atau masukan adalah berupa matriks dua dimensi. *Output* dari masing-masing tahap adalah *feature map* hasil pengolahan dari semua lokasi pada citra masukan. Masing-masing tahap terdiri dari tiga lapisan yaitu konvolusi, aktivasi dan *pooling*. Convolutional Neural Networks (CNN) menggabungkan neural networks dalam dan konvolusi kernel. Convolutional neural network

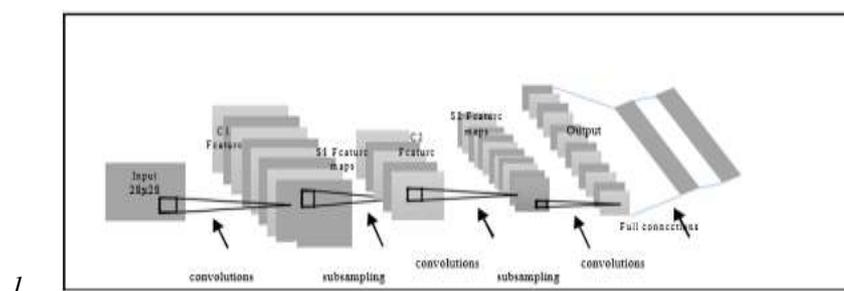
merupakan salah satu model deep learning yang banyak digunakan untuk keperluan analisis citra / visual cortex pada mamalia. CNN memiliki neuron neuron yang disusun secara tiga dimensi, jadi memiliki panjang, lebar, dan tinggi. Sehingga sangat efektif untuk menganalisis gambar. Dalam pembelajaran yang mendalam, neural networks convolutional (CNN, atau ConvNet) adalah kelas neural networks yang mendalam, paling umum diterapkan untuk menganalisis citra visual. CNN menggunakan variasi perceptron multilayer yang dirancang untuk membutuhkan preprocessing minimal. (1) Juga dikenal sebagai shift invariant atau space neural networks neural networks (SIANN), berdasarkan arsitektur bobot bersama dan karakteristik invarian. Jaringan konvolusional diilhami oleh proses biologis (2) dalam bahwa pola konektivitas antara neuron menyerupai organisasi korteks visual hewan. Neuron kortikal individu merespons rangsangan hanya di daerah terbatas bidang visual yang dikenal sebagai bidang reseptif. Bidang reseptif dari neuron yang berbeda sebagian tumpang tindih sehingga menutupi seluruh bidang visual.

CNN menggunakan pra-pemrosesan yang relatif sedikit dibandingkan dengan algoritma klasifikasi gambar lainnya. Ini berarti bahwa jaringan mempelajari filter yang dalam algoritma tradisional direkayasa oleh tangan. Kemandirian ini dari pengetahuan sebelumnya dan upaya manusia dalam desain fitur adalah keuntungan utama. Anda harus memahami bagaimana ide ruang fitur telah muncul. Para peneliti mengetahui neural networks tiruan sebagai penentu fungsi universal dan sejak awal diketahui bahwa beberapa transformasi nonlinier menghaluskan ketidakcocokan dari setiap pemetaan atau pembelajaran. Tetapi masalahnya adalah bagaimana melatih jaringan yang sangat dalam, terutama ketika dimensi ruang input sangat tinggi. Jadi, konsep ruang fitur telah datang, di mana para peneliti secara manual mengurangi jumlah dimensi input dan memuluskan beberapa ketidakcocokan melalui ruang fitur. Tetapi sejak dekade terakhir, karena ketersediaan data tingkat yang cukup, sumber daya komputasi, skema regularisasi, dan penggunaan arsitektur jaringan yang khusus, pelatihan jaringan yang mendalam dimungkinkan. Kemungkinan ini disebut pembelajaran mendalam dan sekarang memengaruhi setiap aspek kehidupan kita. Akhirnya, diamati bahwa representasi data dalam lapisan menengah dari jaringan terlatih mengikuti beberapa pola dari yang sederhana hingga kompleks, yang mirip dengan ruang fitur seperti yang dibahas di atas. Jawaban ini untuk neural networks umpan maju generic Convolutional Neural Network (CNN) terdiri dari satu atau lebih lapisan konvolusional (seringkali dengan langkah subsampling) dan kemudian diikuti oleh satu atau lebih lapisan yang sepenuhnya terhubung seperti pada neural networks multilayer standar. Arsitektur CNN dirancang untuk memanfaatkan struktur 2D dari gambar input (atau input 2D lainnya seperti sinyal ucapan). Ini dicapai dengan koneksi lokal dan bobot terikat diikuti oleh beberapa bentuk penggabungan yang menghasilkan fitur invarian.

Manfaat lain dari CNN adalah mereka lebih mudah dilatih dan memiliki banyak parameter lebih sedikit daripada jaringan yang terhubung penuh dengan jumlah unit tersembunyi yang sama. Arsitektur CNN dan algoritma back propagation untuk menghitung gradien sehubungan dengan parameter model untuk menggunakan optimasi berbasis gradien. [7]

## 2.1 Arsitektur CNN

Secara umum, arsitektur dari sebuah *convolution network* ditunjukkan pada **Gambar 2.12** sebagaimana digunakan oleh LeCun [7] Pada gambar tersebut, *Input* dari CNN adalah berupa citra dengan ukuran tertentu. Tahap pertama dalam CNN adalah tahap konvolusi. Konvolusi dilakukan dengan menggunakan kernel dengan ukuran tertentu. Jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. *Output* dari tahap ini kemudian dikenakan fungsi aktivasi, yang bisa berupa fungsi tanh atau *Rectifier Linear Unit* (ReLU). *Output* dari fungsi aktivasi kemudian melalui proses *sampling* atau *pooling*. *Output* dari proses *pooling* adalah citra yang telah berkurang ukurannya, tergantung dari *pooling mask* yang dipakai.



2. Gambar 2. 1 Struktur Convolutional Neural Network

### 2.2. Convolutions Layer

Convolution Layer merupakan blok bangunan inti CNN, dimana sebagian komputasi dilakukan di lapis ini. Convolution layer dengan satu lembar neuron berisi 28 x 28. Masing masing terhubung dengan suatu area kecil dalam (citra) masukan, masukkan 5 x 5 (piksel yang merupakan bidang reseptif (receptive field) untuk setiap neuron dan menyatakan bahwa filter yang digunakan berukuran 5 x 5, seluruh bidang reseptif akan ditelusuri secara tumpang tindih parsial, maka semua neuron tersebut pasti berbagi bobot koneksi (weight sharing)

Convolution layer dalam arsitektur CNN umumnya menggunakan lebih dari satu filter. Jika menggunakan empat filter maka convolution layer [8]

Neuron ke (i,j) pada hidden layer, memiliki nilai keaktifan y yang dihitung sesuai dengan Persamaan (3.7), dimana nilai (m,n) pada persamaan tersebut menunjukkan ukuran local receptive fields/kernel.

$$y_{i,j} = \sigma(b + (w * x)) \quad (3.7)$$

$$N * x)_{i,j} = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} w_{k,l} x_{i+k,j+l} \quad (3.8)$$

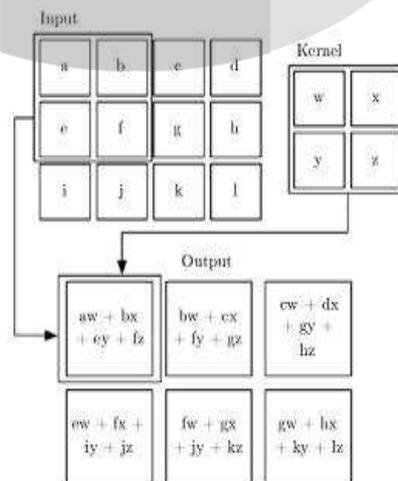
Perkalian antara input dengan kernel di atas (Persamaan (3.8)) yang biasa disebut konvolusi. Namun, menurut Goodfellow, dkk. (2016) konvolusi dilakukan pada kernel yang terbalik, seperti pada Persamaan (3.9). Sedangkan jika kernel tidak dibalik maka fungsi itu disebut cross-correlation. Walaupun begitu, banyak kode pustaka machine learning yang menggunakan rumus cross-correlation dan menyebutnya sebagai rumus konvolusi.

$$(W * x)_{i,j} = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} w_{k,l} x_{i-j,k-l} \quad (3.9)$$

$$=0 \quad =0 \quad , \quad - \quad , \quad -$$

Ukuran citra hasil konvolusi berkurang dibandingkan dengan citra awal dan dapat dinyatakan dengan Persamaan (3.10). Dalam hal ini jika citra dengan ukuran 28x28 dikenai konvolusi dengan ukuran kernel 3x3 maka ukuran akhir menjadi 28-3+1 x 28-3+1 =26x26.

$$ukuran\ hasil\ konvolusi = ukuran\ awal - filter\ size + 1 \quad (3.10)$$



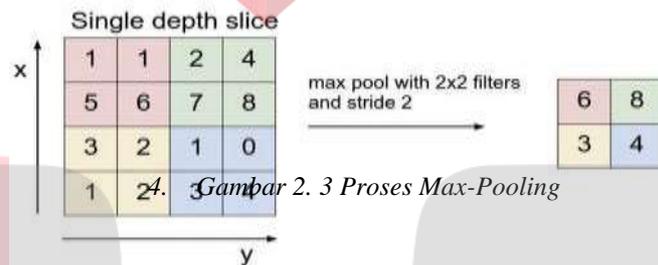
3. Gambar 2. 2 Proses konvolusi pada input array 2D dengan bobot 2D [9]

Pada Gambar 2.13 diatas menunjukkan ilustrasi proses konvolusi pada citra, yang merupakan array dua dimensi I, dengan bobot K (dua dimensi). Pada gambar tersebut, citra berukuran 4x3 dikonvolusi dengan menggunakan kernel berukuran 2x2. Citra yang dihasilkan adalah berukuran 3x2. Elemen pertama pada citra hasil konvolusi

adalah merupakan jumlah dari perkalian bobot kernel dengan nilai citra yang bersangkutan.

**2.3. Pooling Layer**

*Pooling* atau *subsampling* adalah pengurangan ukuran matriks dengan menggunakan operasi *pooling*. Terdapat dua macam *pooling* yang sering dipakai yaitu *average pooling* dan *max - pooling*. Dalam *average pooling*, nilai yang diambil adalah nilai rata-rata, sementara pada *max pooling*, nilai yang diambil adalah nilai maksimal. **Gambar 3.10** menunjukkan operasi *max - pooling*.

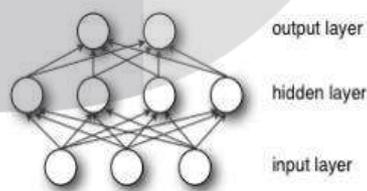


Gambar 2. 3 Proses Max-Pooling

*Output* dari proses *pooling* adalah matriks dengan dimensi yang lebih kecil dibanding dengan matriks awal. Proses konvolusi dan *pooling* dilakukan beberapa kali sehingga didapatkan peta fitur dengan ukuran yang dikehendaki. Peta fitur tersebut akan menjadi input bagi *fully connected neural network*. [8]

**2.4. Multi Layer Neural Network**

Hasil dari *layer* konvolusi (konvolusi, aktivasi, pooling) menjadi *input* bagi *multi-layer perceptron neural network*. Gambar 3.17 menunjukkan *neural network* dengan satu lapisan tersembunyi. Pada gambar tersebut, *neural network* dengan satu lapisan tersembunyi pada gambar tersebut bisa dinyatakan sebagai fungsi :  $\rightarrow$ , dimana  $D$  adalah ukuran dari vektor input, dan  $h$  adalah ukuran dari output vector ( $h$ ). Fungsi tersebut dapat ditulis dalam notasi matriks sebagai  $o(x) = (b^{(2)} + W^{(2)} * (b^{(1)} + W^{(1)} * x))$ , dimana  $b^{(1)}$  dan  $b^{(2)}$  adalah nilai bias,  $W^{(1)}$  dan  $W^{(2)}$  adalah matriks bobot, serta  $\sigma$  adalah fungsi aktivasi.



5. Gambar 2. 4 Neural Network Satu Lapisan Tersembunyi  $o(x) = (b^{(2)} +$

Vektor  $h(x)$  yang merepresentasikan fungsi pada lapisan tersembunyi bisa dinyatakan sebagai  $h(x) = (b^{(1)} + W^{(1)} * x)$ , dimana  $W^{(1)} \in h$  adalah matriks bobot yang menghubungkan vektor input ke lapisan tersembunyi. Sementara fungsi aktivasi  $\sigma$  bisa berupa fungsi tanh atau fungsi sigmoid.

Vektor output dapat dinyatakan sebagai

$$\sigma(h(x)).$$

Dalam *CNN*, *multi layer perceptron* ini diterapkan pada *fully connected layer*. Klasifikasi output dilakukan dengan menggunakan fungsi *softmax*. [10]

## 2.5. Softmax Layer

Fungsi *softmax* digunakan untuk metode klasifikasi dengan jumlah kelas yang banyak, seperti regresi logistik multinomial, analisis diskriminan linear *multiclass*, *Naive Bayes Classifier*, dan *Artificial Neural Network* (ANN). *Softmax* adalah sebuah fungsi yang mengubah K-dimensi vektor 'x' yang berupa nilai sebenarnya menjadi vektor dengan bentuk yang sama namun dengan nilai dalam rentang 0-1, yang jumlahnya 1. Fungsi *softmax* digunakan dalam *layer* yang terdapat pada *neural network* dan biasanya terdapat pada *layer* terakhir untuk mendapatkan *output*. Tidak jauh berbeda dengan neuron pada umumnya *softmax* neuron menerima input lalu melakukan pembobotan dan penambahan bias. Tetapi setelah itu neuron pada *softmax layer* tidak menerapkan fungsi aktivasi melainkan menggunakan fungsi *softmax*.

Jika diketahui  $p$  adalah input berbobot yang diterima oleh neuron pada *softmax layer* maka aktivasi  $y_i$  untuk neuron ke- $i$  adalah:

$$y_i = \frac{e^{p_i}}{\sum_{j=1}^K e^{p_j}} \quad (3.11)$$

Di mana bagian penyebut pada Persamaan (3.11) merupakan total nilai masing-masing neuron pada output layer. Sehingga bisa dikatakan pada *softmax layer*, *output* merupakan distribusi probabilitas untuk setiap kelas. Penyebutnya memastikan bahwa output ke- $i$  berjumlah mendekati 1. Dengan menggunakan *softmax* kita bisa menafsirkan *output* jaringan sebagai perkiraan. [11]

## 2.6. Augmentasi Data

Augmentasi data digunakan selama proses pelatihan CNN. Terutama ketika himpunan citra latih relatif sedikit. Augmentasi data ini dapat dipandang seperti ensemble learning, di mana sejumlah keluaran CNN dapat dilakukan voting (jika CNN berupa model klasifikasi) atau dirata-rata (jika CNN berupa model regresi) untuk menghasilkan keluaran akhir CNN tersebut (Lee et al. 2014). Teknik augmentasi data dapat digunakan dengan sejumlah cara, diantaranya adalah flip, random cropping, color jitter dan random combination. Flip atau pembalikan citra dapat dilakukan secara horizontal dan vertikal. Misalnya citra wajah yang menghadap ke kiri dapat dibalik menjadi ke kanan. Random Cropping untuk membuat variasi data dari berbagai sudut pandang. Seolah kita menguji sebuah masukan dengan banyak model klasifikasi.

Color jitter untuk mendapatkan variasi warna yang acak. Cara paling sederhana dalam teknik ini adalah dengan melakukan variasi kontras secara acak. Cara lain yang lebih kompleks adalah dengan mengaplikasikan principal component analysis (PCA) terhadap semua komponen warna RGB piksel dalam citra latih. Kemudian, lakukan penyempitan "himpunan warna" sepanjang arah principal component. Terakhir, tambahkan offset ke dalam semua piksel pada suatu citra latih. Random Combination untuk sejumlah teknik pengolahan citra seperti translation atau pembesaran/pengecilan, rotation atau pemutaran dengan derajat tertentu, stretching atau peregangan/pengerutan, shearing, dan lens distortions. Pada saat pelatihan dapat dilakukan augmentasi data setiap data hasil augmentasi dilatihkan ke CNN dengan label yang sama dengan citra latih aslinya. Citra dilatih secara berulang-ulang sampai nilai loss yang relatif kecil, yang berarti CNN sudah bisa membedakan kedua citra tersebut secara akurat. Setelah proses pelatihan selesai, CNN diuji. Citra diuji dengan augmentasi dengan sejumlah teknik seperti pada saat pelatihan. Setiap citra hasil augmentasi dimasukkan ke CNN sedemikian hingga dihasilkan sejumlah label hasil pengenalan. Selanjutnya, kita melakukan voting untuk menentukan label keputusan final CNN [12].

## 2.7. Ukuran Performansi

Untuk mengevaluasi model pembelajaran sehingga dapat memilih satu yang terbaik. Confusion matrix adalah matriks yang berisi informasi tentang hasil prediksi klasifikasi dan data aktual yang dilakukan oleh sistem klasifikasi dan data aktual yang dilakukan oleh sistem klasifikasi. Untuk mengevaluasi suatu model klasifikasi, kita memerlukan himpunan data uji yang tidak digunakan dalam pelatihan. Evaluasi dapat dilakukan menggunakan suatu ukuran tertentu, tujuh diantaranya.

TP adalah true positives, jumlah tuple positif yang dilabeli dengan benar oleh model klasifikasi. Yang dimaksud tuple positif adalah tuple aktual yang berlabel positif, seperti tuple dengan label layak = 'Ya'

TN adalah true negatives, jumlah tuple negatif yang dilabeli dengan benar oleh model klasifikasi. Yang dimaksud tuple negatif adalah tuple aktual yang berlabel negatif, seperti tuple dengan label layak = 'Tidak'

FN adalah false negatives, jumlah tuple positif yang salah dilabeli oleh model klasifikasi. Misalnya, sebuah tuple pelanggan yang berlabel Layak = 'Ya' tetapi oleh model klasifikasi dilabeli Layak = 'Tidak'.

Confusion matrix sangat berguna untuk menganalisis kualitas model klasifikasi dalam mengenali tuple dari kelas yang ada. TP dan TN menyatakan bahwa model klasifikasi mengenali tuple dengan benar, artinya tuple positif dikenali sebagai positif dan tuple negatif dikenali sebagai negatif. Sebaliknya, FP dan FN menyatakan bahwa model klasifikasi salah dalam mengenali tuple, tuple negatif dikenali sebagai positif dan tuple positif dikenali sebagai negatif. P' adalah jumlah tuple

Accuracy atau tingkat pengenalan menyatakan persentase dari jumlah tuple dalam data uji yang diklasifikasikan dengan benar oleh model klasifikasi.

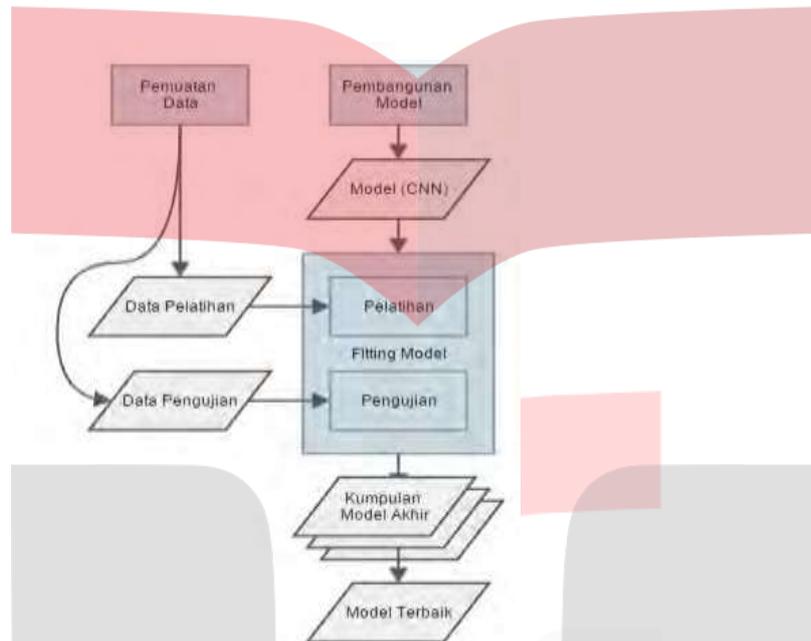
Error rate (tingkat kesalahan) atau misclassification rate yang dihitung menggunakan rumus atau bisa juga dinyatakan sebagai  $1 - \text{accuracy}$ . Dengan rumus tersebut, maka error rate.

Accuracy dan error rate digunakan untuk klasifikasi data dengan kelas seimbang dimana jumlah data pada setiap kelas relatif sama. Untuk klasifikasi data dengan yang tidak seimbang, anda memerlukan ukuran lain yaitu precision dan recall.

1. Precision adalah ukuran kepastian yaitu berapa persentase tuple yang dilabeli sebagai positif dengan benar pada kenyataannya
2. Recall adalah ukuran kelengkapan, yaitu berapa persentase tuple positif yang dilabeli sebagai positif. Dengan demikian, recall adalah sama dengan sensitivity atau true positive rate.
3. Kecepatan proses dapat diukur berdasarkan kompleksitas komputasi secara matematis atau secara empiris berdasarkan eksperimen
4. Ketahanan terhadap derau, atau data dengan atribut bernilai kosong (missing value)

Skalabilitas terhadap ukuran data dapat diukur secara matematis maupun empiris apakah model klasifikasi anda dapat digunakan secara efisien untuk sejumlah data yang berukuran semakin besar dan besar [13]

## 2.8 Skenario Implementasi



Gambar 3. 1 Flowchart Machine Learning

Pada proses pembangunan model CNN, akan diterima sejumlah hyperparameter yang mendefinisikan jaringan CNN yang digunakan, lalu akan dibangun sebuah model CNN awal sesuai input. Model awal tersebut kemudian akan diolah dalam proses *fitting* data dimana parameter pada jaringan (bobot dan bias) akan dikonfigurasi sesuai dengan dataset melalui proses pelatihan. Proses *fitting* data akan menghasilkan beberapa model CNN akhir, lalu akan dipilih model dengan nilai *error* terkecil sebagai model terbaik.

## 2.9. Skenario Preprocessing

### 2.10. Dataset

Dataset terkait jerawat diberi label dikarenakan jenis pembelajaran adalah pembelajaran supervised. Data pelatihan digunakan untuk melatih sebuah algoritma.

Semakin banyak data pelatihan maka proses klasifikasi akan semakin baik [14]

### 2. 11. Dataset Training

Dataset pelatihan adalah kumpulan contoh yang digunakan selama proses pembelajaran dan digunakan agar sesuai dengan parameter (mis., Bobot) dari, misalnya, klasifikasi jerawat. Sebagian besar pendekatan yang mencari melalui data pelatihan untuk hubungan empiris cenderung sesuai dengan data, yang berarti bahwa mereka dapat mengidentifikasi dan mengeksploitasi hubungan nyata dalam data pelatihan yang tidak berlaku secara umum [14]

### 2.12. Dataset Testing

Dataset uji adalah dataset yang tidak tergantung pada dataset pelatihan, tetapi yang mengikuti distribusi probabilitas yang sama dengan dataset pelatihan. Jika model cocok dengan dataset pelatihan juga cocok dengan dataset uji dengan baik, overfitting minimal telah terjadi (lihat gambar di bawah). Pemasangan yang lebih baik dari dataset pelatihan dibandingkan dengan dataset tes biasanya menunjuk pada overfitting. [14]

### 2.13. Dataset Validasi

Dataset validasi masih bagian dari training set tetapi tidak terlibat dalam pembuatannya. Dataset validasi bisa untuk mengecek overfitting atau perubahan nilai Hyperparameter. [15]

#### 2.14. Skenario pemilihan model beserta optimizernya

Skenario pemilihan model pada perancangan CNN dalam penelitian ini adalah sebagai berikut: Pada CNN, jenis layer lebih bervariasi dibandingkan MLP sehingga arsitektur jaringan memiliki banyak variasi. Ukuran jaringan (kedalaman dan lebar) adalah hyperparameter jaringan yang perlu di tuning terhadap dataset yang digunakan. Dikarenakan proses tuning ukuran jaringan yang sangat lama, pada tugas akhir ini kedalaman akan dibatasi maksimal lima convolution layer dan lebar jaringan akan mengambil nilai awal dari All Convolutional Nets yaitu 96 neuron, lalu akan dilakukan eksplorasi pada nilai kelipatan 96. Pada CNN, terdapat dua bagian utama dari arsitektur model, bagian ekstraksi fitur, dan bagian klasifikasi. Bagian ekstraksi fitur berfungsi untuk mentransformasi data input sehingga dapat diklasifikasi dengan baik. Hal tersebut dilaksanakan dengan mempropagasi data melalui berbagai transformasi linear maupun *non-linear*. Bagian klasifikasi berfungsi untuk mereduksi data sehingga dapat dimasukkan ke dalam softmax classifier untuk dilakukan klasifikasi.

#### 2.15. Skenario Pengujian

Skenario pengujian menggunakan beberapa arsitektur model dan data set seperti Mini VGGnet versus beberapa dataset (Dermnet, Dermnet Crop, Acnesmall). Disamping itu dilakukan pula terkait dengan pengujian optimizer SGD dan Adam.

### 3. Pembahasan

Pada bagian pembahasan ini akan dibahas beberapa peningkatan untuk beberapa dataset namun sebagian besar terjadi peningkatan setelah dilakukan augmentasi image. Hasil yang lebih buruk mungkin terjadi karena augmentasi dilakukan agar setiap kelas memiliki jumlah image yang setara (sebelumnya ada kelas yang memiliki ratusan image, namun ada kelas lain yang hanya terdiri atas beberapa gambar) sehingga ada kelas yang mungkin jumlah augmentasinya kurang lebih sama dengan dataset asli. Sedangkan, untuk dataset asli yang jumlahnya hanya sedikit, hasil training menunjukkan peningkatan yang signifikan. Hal ini tidak dapat menjadi bahan simpulan bahwa sistem sudah cukup bagus untuk melakukan klasifikasi karena bisa jadi data validasi dan data training berasal dari image yang sama.

#### 3.1. Kerangka Convolutional Neural Network (CNN)

Performa dari sebuah CNN sangat bergantung pada arsitektur jaringan. Pada tugas akhir ini dibandingkan arsitektur yang dibangun berdasarkan beberapa referensi. Lebar layer pada jaringan dibuat incremental mengikuti kelipatan  $K$ , dimana  $K$  adalah variabel ketebalan layer pada jaringan ( $k$  size). Dalam mendeskripsikan arsitektur jaringan, akan digunakan notasi arsitektur CNN yang mendeskripsikan jenis, ukuran, dan urutan layer yang digunakan pada arsitektur yang bersangkutan. Notasi disusun dengan urutan dari kiri ke kanan dan menggunakan karakter sebagai pemisah antar layer.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 32)	896
activation (Activation)	(None, 64, 64, 32)	0
batch_normalization (Batch Normalization)	(None, 64, 64, 32)	128
conv2d_1 (Conv2D)	(None, 64, 64, 32)	9248
activation_1 (Activation)	(None, 64, 64, 32)	0
batch_normalization_1 (Batch Normalization)	(None, 64, 64, 32)	128
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
activation_2 (Activation)	(None, 32, 32, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 32, 32, 64)	256
conv2d_3 (Conv2D)	(None, 32, 32, 64)	36928
activation_3 (Activation)	(None, 32, 32, 64)	0
batch_normalization_3 (Batch Normalization)	(None, 32, 32, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_1 (Dropout)	(None, 16, 16, 64)	0
Flatten (Flatten)	(None, 16384)	0
dense (Dense)	(None, 512)	8389120
activation_4 (Activation)	(None, 512)	0
batch_normalization_4 (Batch Normalization)	(None, 512)	2048
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 5)	2565
activation_5 (Activation)	(None, 5)	0

Total params: 8,460,069  
 Trainable params: 8,458,661  
 Non-trainable params: 1,408

Gambar 4. 1 Parameter Neural Network

Tabel 4. 1 number of neuron, input, output and hidden layer

Neuron node	Input	Output	Hidden layer	Neuron in hidden layer
16384	64x64 pixels	Dense and activation5	Dense and activation4	521

3.2. Rekapitulasi hasil training dan validasi sistem.

Tabel 4. 2 Rekapitulasi hasil training dan validasi system

Dataset	Augmentasi	Rasio augmentasi	Optimizer	Macro-Average	Weighted-Avg
Dermnet	None (433 images)	1	SGD	0.42	0.41
	Augmented on-the-fly	1	SGD	0.30	0.31
	Augmented Manual 1 (745 images)	1.7	SGD	0.52	0.52
	Augmented Manual 2 (2566 images)	5.9	SGD	0.64	0.64
Dermnet Crop (menghilangkan watermark pada gambar)	Pure Dataset (866 images)	1	SGD	0.44	0.44
	Augmented on-the-fly	1	SGD	0.48	0.48

	Augmented Manual 1 (914 images)	1.0	SGD	0.44	0.44
	Augmented Manual 2 (2786 images)	3.2	SGD	0.37	0.37
Acne Small (Training Data)	Pure Dataset (100 images)	1	SGD	0.36	0.38
	Augmented on-the-fly	1	SGD	0.33	0.38
	Augmented Manual 1 (572 images)	5.72	SGD	0.79	0.81
	Augmented Manual 2 (2461 images)	24.6	SGD	0.98	0.98

**Train Model 1**

Pada Model 1, akurasi yang di dapat 0,733 dan kurva validation loss (val\_loss) terdapat spike yang menunjukkan bahwa loss function pada validation loss tidak berkurang secara sempurna, karena semakin rendah nilai loss (menuju nol) maka semakin sempurna sebuah model.

**Train Model 2**

Pada Model 2 , akurasi yang di dapat 0,9333 dan model ini menunjukkan kurva yang terbaik dibandingkan model 1 dan 3, walaupun ada spike di awal namun kurva val\_loss menuju nol asimptotik. Dan kurva train\_acc menuju satu.

**Train Model 3**

Pada Model 3, akurasi yang di dapat adalah 0.85 dan model ini menunjukkan kurva yang kurang sempurna, terdapat banyak spike pada kurva val\_loss. Dan kurva train\_acc menuju satu walau kurang smooth.

**3.3. Hasil**

Model: Model 1, Test Set: acnesmall\_split/test/



6. *Gambar 4. 2 Model 1 acnesmall\_split/test/*

Model: Model 2, Test Set: acnesmall\_split/test/



7. *Gambar 4. 3 Model 2 acnesmall\_split/test*

Model: Model 3, Test Set: acnesmall\_split/test/



8. *Gambar 4. 4 Model 3 acnesmall\_split/test/*

Model: Model 1, Test Set: acnesmall-jpg/



9. *Gambar 4. 5 Model 1 acnesmall-jpg*

Model: Model 2, Test Set: acnesmall-jpg/



10. *Gambar 4. 6 Model 2 acnesmall-jpg/*

Model: Model 3, Test Set: acnesmall-jpg/



11. *Gambar 4. 7 Model 3 acnesmall-jpg/*

#### 4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan makadapat diambil kesimpulan sebagai berikut: Implementasi struktur jaringan dapat mengklasifikasi jenis-jenis jerawat dengan menggunakan Convolutional Neural Network (CNN) dengan penjelasan berikut:

1. Citra dapat melewati proses *preprocessing* dengan baik yaitu dengan diubah ukuran dimensinya menjadi 46x46 dan dikonversi citra warnanya menjadi *greyscale*.
2. Akurasi dapat dicapai semakin baik apabila digunakan data *train* yang semakin besar. Hal ini dibuktikan dengan pengujian pada tiga jumlah data *training* berbeda, yaitu 70 %, 80 % dan 90 %. Dari ketiganya, hasil akurasi terbaik pada data uji dihasilkan oleh data *train* sebesar 90 % dan data *test* sebanyak 10 %.
3. Dengan menggunakan data *train* 90 % dan data *test* 10 %, dua lapisan konvolusi, jumlah filter 10 pada lapisan konvolusi pertama dan jumlah filter 20 pada lapisan konvolusi kedua, ukuran kernel pada lapisan konvolusi pertama dan kedua sebesar 3x3 diperoleh akurasi model dalam mengenali berdasarkan data uji telah sangat baik yaitu sebesar 91.6 %.

Sesuai hasil penelitian ini, terdapat beberapa saran bagi penelitian selanjutnya khususnya bagi penelitian yang menggunakan metode *Convolutional Neural Network*:

1. Guna dapat mempercepat pemrosesan pelatihan CNN, dapat digunakan perangkat *Graphical Processing Unit* (GPU).
2. Dapat mengembangkan program menjadi lebih *attractive*, contohnya dengan menggabungkan dengan ilmu *computer vision*

#### Reference

- [1] Hay, R., Johns, N., Williams, H., Bolliger, I., Dellavalle, R., Margolis, D., Naghavi, M. The Global Burden of Skin Disease in 2010: An Analysis of the Prevalence and Impact of Skin Conditions. *Journal of Investigative Dermatology*, 1527-1534 (2013).
- [2] Hafsah, I., & Andono, P. Deteksi Otomatis Penyakit Kulit Menggunakan Algoritma Naive Bayes (2015)
- [3] Chang, C.-Y., & Liao, H.-Y.. Automatic Facial Spots and Acnes Detection System. *Journal of Cosmetics, Dermatological Sciences and Applications*, 3, 28-35. (2013)
- [4]. Liu, Z., & Zerubia, J. Towards Automatic Acne Detection Using a MRF Model with Chromophore Descriptors. *European Signal Processing Conference (EUSIPCO)*. Marrakech: IEEE. (2013)
- [5] Ramli, R., Malik, A., Hani, A., & Yap, F.-B. Identification of Acne Lesions, Scars and Normal Skin for Acne Vulgaris Cases. *National Postgraduate Conference (NPC)*. Kuala Lumpur: IEEE (2011)
- [6] Doneva, C., Hidayat, B., & Rizal, A. Identifikasi Jenis Jerawat Berdasarkan Kombinasi Segmentasi Warna dan Analisis Tekstur dengan Deteksi Binary Large Object (BLOB) Menggunakan Jaringan Syaraf Tiruan-Learning Vector Quantization. (2013)
- [7] LeCun, Y., Kavukcuoglu, K. dan Farabet, C., Convolutional Networks and Applications in Vision, ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems, pp.253–256. 2010.
- [8] Steve Lawrence, Lee Giles, Ah Chung Tsoi, Andrew D. Back, Face Recognition: A Convolutional Neural-Network Approach, *NEURAL NETWORKS*, VOL. 8, NO. 1, IEEE, 1, JANUARY 1997.
- [9] Rismiyati, Implementasi Convolutional Neural Network Untuk Sortasi Mutu Salak Ekspor Berbasis Citra Digital, Tesis, Program Ilmu Komputer, Universitas Gajah Mada, Yogyakarta. 2016.
- [10] Patrice Y. Simard, Dave Steinkraus, John C. Platt, Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, Microsoft Research, One Microsoft Way, 2002
- [11] W Luo, Y Li, R Urtasun, R Zemel Understanding the Effective Receptive Field in Deep Convolutional Neural Networks, *Advance in Neural Information*, 2016.

- [12] J Sanjaya, M Ayub, Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup, Jurnal Teknik informatika dan Sistem, 2020.
- [13] A Santoso, G Ariyanto, Implementasi Deep Learning Berbasis Keras untuk Pengenalan Wajah, Jurnal Teknik Elektro, 2018.
- [14] Shah Tarang, About Train, Validation and test set in Machine Learning, TarahShah.com.2017
- [15] Sanjay M, Why and How to Cross Validate a Model, Towardsdatascience.com.2018.