Penerapan *Attention Module* pada *MobileNetV3* untuk Klasifikasi Penyakit Kulit

Mohammad Alif Tuharea[1)], Nahumi Nugrahaningsih[2)], Rony Teguh[3)]

[1)2)3)]Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Kampus Tunjung Nyaho Jalan Yos Sudarso, Palangka Raya, Kalimantan Tengah, Indonesia
[1)]alif.tuharea04@gmail.com
[2)]nahumi@it.upr.ac.id
[3)]ronyteguh@it.upr.ac.id

Abstrak

Penyakit kulit merupakan masalah kesehatan global yang signifikan. Berdasarkan data *International League of Dermatological Societies* (ILDS), terdapat 4,69 miliar kasus pada tahun 2021 yang menjadikannya penyebab keempat beban non-fatal terbesar di dunia. Diagnosis tradisional sangat bergantung pada pengalaman dan subjektivitas tenaga medis, sehingga diperlukan pendekatan berbasis komputasi cerdas.

Penelitian ini mengusulkan integrasi *Convolutional Block Attention Module* (CBAM) pada arsitektur MobileNetV3 untuk meningkatkan akurasi klasifikasi penyakit kulit. Tiga konfigurasi diuji: (a) CBAM setelah *Inverted Residual Block* dengan *SE Block* di dalamnya (CBAM A), (b). CBAM setelah *SE Block* (CBAM B), dan (c) CBAM menggantikan *SE Block*. Dataset yang digunakan merupakan gabungan tiga dataset publik dengan total 17.467 gambar dari 28 kelas penyakit dengan distribusi tidak seimbang. Model dilatih menggunakan *transfer learning* dengan pembekuan *backbone*, dievaluasi pada data uji tidak seimbang melalui lima kali pengulangan *seed* berbeda. Performa model dievaluasi dengan metrik *accuracy*, *precision*, *recall*, dan *F1-Score*. Selain itu, model dianalisis dengan uji-t berpasangan.

Hasil eksperimen menunjukkan bahwa penempatan CBAM setelah *SE Block* (CBAM B) menghasilkan peningkatan performa tertinggi secara signifikan (berdasarkan paired t-test dan uji signifikansi statistik) dengan akurasi 0,4768 ± 0,0133, precision 0,4635 ± 0,0168, recall 0,4898 ± 0,0136, dan F1-Score 0,4721 ± 0,0156. Konfigurasi CBAM A justru menurunkan performa secara drastis, sedangkan CBAM C menghasilkan performa di bawah baseline. Penelitian ini menegaskan bahwa integrasi CBAM dapat meningkatkan performa model MobileNetV3 dengan pertimbangan bahwa posisi integrasi sangat mempengaruhi efektivitasnya dalam arsitektur CNN ringan.

Kata kunci: Penyakit kulit, MobileNetV3, Convolutional Block Attention Module (CBAM), Integrasi

Abstract

Skin diseases constitute a significant global health burden. The International League of Dermatological Societies (ILDS) reports 4.69 billion cases in 2021, making them the fourth leading cause of non-fatal disability worldwide. Traditional diagnosis depends heavily on clinician experience and subjectivity, motivating the development of intelligent computational approaches.

This study proposes integrating the Convolutional Block Attention Module (CBAM) into the MobileNetV3 architecture to improve skin disease classification accuracy. Three integration configurations are examined: a) CBAM placed after an Inverted Residual Block that already contains an SE block (CBAM A), b) CBAM inserted immediately after the SE block (CBAM B), and c) CBAM replacing the SE block entirely (CBAM C). A merged dataset of 17,467 images

DOI:https://doi.org/10.47111/jointecoms.v6i3 *Received: 01-09-2026*
Accepted: 10-09-2026

from 28 disease classes with an imbalanced distribution is constructed from three public sources. Models are trained with transfer learning while freezing the backbone and evaluated on an unseen test set across five different random seeds. Performance is measured using accuracy, precision, recall, and F1-score, and statistical significance is assessed with a paired t-test.

Experimental results demonstrate that placing CBAM after the SE block (CBAM B) yields the highest improvement, significantly outperforming the baseline ($p < 0.05$) with accuracy 0.4768 ± 0.0133, precision 0.4635 ± 0.0168, recall 0.4898 ± 0.0136, and F1-score 0.4721 ± 0.0156. In contrast, CBAM A drastically degrades performance, and CBAM C falls below the baseline. These findings confirm that integrating CBAM can enhance MobileNetV3 for skin disease classification, provided that the integration position is carefully chosen, as it strongly influences the effectiveness of attention mechanisms in lightweight CNN architectures.

Keywords: *Skin disease, MobileNetV3, Convolutional Block Attention Module (CBAM), Integration*

1. PENDAHULUAN

Penyakit kulit merupakan salah satu permasalahan yang banyak terjadi di masyarakat di mana berdasarkan data dari *International League of Dermatological Societies* (ILDS) mencatat terdapat 4,69 miliar kasus pada tahun 2021 yang menjadikannya sebagai penyebab keempat beban non-fatal terbesar di dunia dan masuk dalam 10 penyebab utama disabilitas. Penyakit ini sendiri tidak hanya berdampak pada kondisi fisik, tetapi juga dapat menurunkan kualitas hidup penderitanya apabila tidak terdeteksi dan ditangani secara tepat sehingga diperlukan Upaya deteksi dini yang akurat dan efisien [1].

Penyakit kulit cenderung memiliki variasi visual yang tinggi dan sering kali menunjukkan perbedaan karakteristik yang bersifat halus antar kelas penyakit. Proses diagnosis secara tradisional sangat bergantung pada pengalaman dan subjektivitas tenaga medis. Perbedaan Tingkat keahlian, serta kemiripan visual antar jenis penyakit kulit tentunya menjadi tantangan tersendiri dalam penanganan penyakit kulit. Permasalahan ini mendorong berkembangnya pendekatan berbasis komputasi cerdas, khususnya *machine learning* dan *deep learning* untuk membantu proses klasifikasi penyakit kulit secara otomatis dan objektif [2].

Dalam beberapa tahun terakhir, pendekatan berbasis *deep learning*, khususnya *Convolutional Neural Network* (CNN), telah banyak dimanfaatkan untuk membantu proses klasifikasi penyakit kulit berbasis gambar. CNN mampu mempelajari pola tekstur, warna, dan bentuk lesi kulit tanpa memerlukan rekayasa fitur manual, sehingga banyak digunakan dalam sistem klasifikasi penyakit kulit berbasis citra [3]. Namun demikian,arsitektur CNN konvensional umumnya memiliki jumlah parameter yang besar dan kebutuhan komputasi tinggi, sehingga kurang ideal untuk diterapkan padaperangkat dengan sumber daya terbatas atau sistem *real-time*.

Sebagai Solusi atas permasalahan tersebut, arsitektur CNN ringan seperti MobileNet dikembangkan dengan mengutamakan efisiensi komputasi tanpa mengorbankan performa secara signifikan. MobileNetV3 merupakan pengembangan lanjutan dari MobileNet yang dirancang untuk tetap akurat dengan beban komputasi yang jauh lebih ringan dibandingkan arsitektur lain seperti ResNet atau EfficientNet, sehingga arsitektur ini cocok untuk digunakan pada lingkungan yang memiliki keterbatasan biaya komputasi [4].

Meskipun demikian, MobileNetV3 masih memiliki keterbatasan dalam menangkap fitur-fitur penting yang bersifat kompleks dan berskala kecil pada citra penyakit kulit. Ciri visual penyakit kulit sering kali memiliki variasi bentuk, ukuran, dan pola halus, sehingga model dapat kehilangan informasi penting apabila tidak diarahkan untuk fokus pada area dan kanal fitur yang relevan. Untuk mengatasi hal tersebut, mekanisme attention diperkenalkan sebagai pendekatan untuk meningkatkan kemampuan CNN dalam memfokuskan perhatian pada fitur yang paling informatif.

Penelitian terbaru menunjukkan bahwa penerapan mekanisme attention pada arsitektur CNN ringan mampu meningkatkan performa klasifikasi tanpa menambah beban komputasi secara signifikan. Integrasi modul attention seperti CBAM (Convolutional Block Attention Module) ke dalam CNN telah terbukti meningkatkan akurasi dan generalisasi model pada berbagai kasus klasifikasi citra medis [5].

2. TINJAUAN PUSTAKA

2.1 Penyakit Kulit

Penyakit kulit adalah kelainan kulit akibat adanya jamur, kuman, parasite,virus maupun infeksi yang dapat menyerang siapa saja dari segala umur. Penyakit kulit dapat menyerang seluruh maupun bagian tubuh tertentu dan dapat memperburuk kondisi kesehatan penderita jika tidak ditangani secara serius.

Menurut [6], terdapat beberapa faktor yang dapat menyebabkan terjadinya penyakit kulit di antaranya adalah virus, bakteri, sistem imun, dan kebersihan lingkungan.

2.2 Convolutional Neural Network (CNN)

Menurut [7], *Convolutional Neural Network* (CNN) adalah jenis *neural network* yang dirancang khusus untuk menangani masalah pengolahan citra. Arsitektur CNN sendiri terdiri dari 3 lapisan utama yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*.

Beberapa arsitektur yang berbasis CNN adalah MobileNet, GoogleNet, ResNet50, InceptionV3, DenseNet, VGG19, dan NasNet [4].

2.3 MobileNet

MobileNet merupakan salah satu arsitektur CNN yang diperkenalkan oleh Google pada tahun 2015. Arsitektur ini dibangun dengan fokus pada teknik konvolusi *Depthwise Separable Convolution*, dimana pada teknik ini terjadi 2 proses konvolusi yaitu *Depthwise Convolution* dan *Pointwise Convolution*. Pendekatan ini dapat mengurangi beban komputasi sebanyak 9 kali lipat namun tetap mempertahankan tingkat akurasi [8].

Sejak pertama kali diperkenalkan, MobileNet telah mengalami peningkatan versi dengan versi terbaru yaitu MobileNetV4 yang dikenalkan pada September 2024 dengan paper berjudul "MobileNetV4: Universal Models for the Mobile Ecosystem".

2.3.1 MobileNetV3

MobileNetV3 merupakan peningkatan dari arsitektur MobileNetV2 yang dirancang untuk mengoptimalkan keseimbangan antara latensi dan akurasi pada perangkat mobile. Tidak seperti versi sebelumnya yang sepenuhnya bergantung pada desain secara manual, MobileNetV3 menggabungkan *Neural Architecture Search* (NAS) dengan optimisasi jaringan manual. Desain arsitektur dengan pendekatan ini memungkinkan peningkatan kinerja pada CPU mobile.

Model ini membawa beberapa perubahan utama seperti integrasi modul *Squeeze-and-Excitation* (SE) pada beberapa blok *bottleneck*, pengenalan fungsi aktivasi baru yang disebut *hard-swish* yang diterapkan secara selektif pada lapisan yang lebih dalam dari jaringan, dan optimisasi dilakukan pada beberapa lapisan yang mempengaruhi latensi, terutama pada bagian awal dan akhir jaringan [9].

2.4 Attention Module

Menurut [10], *attention module* merupakan modul yang terinspirasi dari cara kerja sensor kognitif manusia yang dapat fokus pada elemen penting sambil menekan informasi yang tidak penting, sehingga dapat meningkatkan akurasi dan efisiensi dalam pengenalan dan pengambilan keputusan.

Dari pemahaman tersebut, dapat disimpulkan bahwa *attention module* merupakan sebuah mekanisme dalam *deep neural network* yang digunakan agar jaringan dapat fokus pada fitur utama dan meredam fitur yang tidak penting untuk meningkatkan akurasi dan efisiensi.

2.4.1 Convolutional Block Attention Module (CBAM)

Convolutional block attention module (CBAM) merupakan *attention module* ringkas untuk *feed forward* CNN dimana modul ini dapat fokus pada informasi penting di *channel* dan spasial, memastikan bahwa modul ini tidak hanya menghemat biaya komputasi namun juga kemudahan integrasinya pada arsitektur jaringan yang ada sebagai modul *plug-and-play* [11].

Berdasarkan paper dari [12], CBAM menggabungkan 2 *attention module* yaitu, *channel attention module* (CAM) dan *spatial attention module* (SAM) secara sekuensial.

CAM bertugas untuk mempelajari fitur 'apa' (channel mana) yang paling penting dengan cara melakukan *global average pooling* dan *global max pooling* pada *feature map* masukan, kemudian memprosesnya melalui *shared multi-layer perceptron* (MLP) dengan rasio reduksi tertentu.

Sementara itu, SAM mempelajari 'di mana' lokasi fitur penting secara spasial dengan menerapkan operasi konvolusi pada *feature map* yang telah diperkaya oleh CAM [13].

2.5 Evaluasi Klasifikasi

2.5.1 Confusion Matrix

Dalam evaluasi model klasifikasi, khususnya pada klasifikasi biner (dua kelas), hasil prediksi model perlu dibandingkan dengan label sebenarnya. Alat yang paling umum digunakan untuk keperluan ini adalah *confusion matrix* (atau disebut juga tabel kontingensi). Confusion matrix pertama kali diperkenalkan oleh Karl Pearson pada tahun 1904 sebagai *contingency table*, dan kemudian dikenal sebagai classification matrix sebelum akhirnya populer dengan istilah *confusion matrix* dalam *data science* [14].

Confusion matrix berbentuk matriks 2x2 yang memuat empat komponen dasar:

- TP (*True Positive*) : jumlah prediksi positif yang benar
- TN (*True Negative*) : jumlah prediksi negatif yang benar
- FP (*False Positive*) : jumlah prediksi positif yang salah
- FN (*False Negative*) : jumlah prediksi negatif yang salah

2.5.2 Akurasi

Akurasi adalah metrik yang paling sederhana dan paling sering digunakan dalam evaluasi klasifikasi. Akurasi didefinisikan sebagai proporsi prediksi yang benar (baik positif maupun negatif) terhadap seluruh prediksi yang dilakukan [14].

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Nilai akurasi berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan bahwa semua prediksi benar, sedangkan nilai 0 menunjukkan semua prediksi salah [15]. Namun, akurasi memiliki keterbatasan yang signifikan ketika digunakan pada data yang tidak seimbang (*imbalanced data*). Oleh karena itu, pada data dengan ketidakseimbangan kelas yang parah, akurasi tidak boleh dijadikan satu- satunya metrik evaluasi [16].

2.5.3 Precision

Precision mengukur seberapa akurat model dalam memprediksi kelas positif. Precision didefinisikan sebagai proporsi prediksi positif yang benar terhadap seluruh prediksi positif yang dibuat oleh model [17].

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Precision berkisar antara 0 hingga 1. Nilai *precision* yang tinggi menunjukkan bahwa model memiliki tingkat *false positive* (FP) yang rendah, artinya ketika model memprediksi positif, kemungkinan besar prediksi tersebut benar.

2.5.4 Recall

Recall juga dikenal sebagai sensitivitas atau *true positive rate* (TPR), mengukur seberapa baik model dalam menemukan semua sampel positif. Recall didefinisikan sebagai proporsi sampel positif yang berhasil diprediksi dengan benar oleh model [18].

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Nilai *recall* berkisar antara 0 hingga 1. Recall yang tinggi menunjukkan bahwa model memiliki tingkat *false negative* (FN) yang rendah, artinya model tidak melewatkan banyak sampel positif.

2.5.5 F1-Score

F1-Score adalah metrik yang menggabungkan *precision* dan *recall* ke dalam satu nilai tunggal dengan menghitung rata-rata harmonik dari keduanya [17].

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision+Recall} \quad (4)$$

Nilai *F1-score* berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan *precision* dan *recall* yang sempurna. Rata- rata harmonik digunakan karena lebih sensitif terhadap nilai yang kecil dibandingkan rata- rata aritmatik, sehingga *F1-score* akan rendah jika salah satu dari *precision* atau *recall* rendah [18].

F1-score sangat berguna ketika kedua jenis kesalahan (FP dan FN) memiliki dampak yang sama pentingnya, atau ketika data tidak seimbang. [14] menekankan bahwa *F1-score* lebih baik daripada akurasi untuk data dengan ketidakseimbangan kelas, karena *F1-Score* mempertimbangkan kinerja pada kelas minoritas. Dalam konteks klasifikasi multi-kelas, terdapat variasi *F1-score* seperti *micro F1*, *macro F1*, dan *weighted F1*, di mana masing- masing memiliki cara agregasi yang berbeda tergantung pada apakah setiap kelas diberi bobot yang sama atau disesuaikan dengan ukuran kelas [18].

2.6 Uji Statistik

Uji statistik merupakan prosedur formal dalam statistika inferensial yang digunakan untuk mengambil keputusan mengenai suatu hipotesis berdasarkan data sampel [19]. Dalam penelitian kuantitatif, uji statistik berfungsi untuk menentukan apakah hasil observasi yang diperoleh kemungkinan besar terjadi secara kebetulan atau mencerminkan efek yang nyata. Berdasarkan asumsi distribusi data, uji statistik dibedakan menjadi statistik parametrik (data berdistribusi normal, skala interval/rasio) dan statistik non-parametrik (data tidak normal atau skala nominal/ordinal) [20].

2.6.1 Paired T-Test

Paired t-test (uji t berpasangan) adalah uji statistik parametrik yang digunakan untuk membandingkan rata-rata dua kelompok data yang saling berpasangan atau berasal dari subjek yang sama. Pasangan data ini dapat berupa pengukuran sebelum dan sesudah perlakuan (*pre-test* dan *post-test*) pada individiu yang sama, atau dua pengukuran pada kondisi berbeda namun dari subjek yang yang sama [21].

Menurut [22], terdapat empat asumsi yang harus dipenuhi:

- Variabel dependen diukur pada skala kontinu (interval atau rasio)
- Variabel independen terdiri dari dua kelompok yang berpasangan
- Tidak terdapat *outlier* (data ekstrem) yang signifikan pada selisih antara kedua kelompok berpasangan.
- Distribusi selisih antara kedua kelmpok berpasangan mendekati normal.

Nilai statistik uji t untuk sampel berpasangan dihitung dengan rumus:

$$t_{hitung} = \frac{\bar{D}}{s_D/\sqrt{n}} \quad (5)$$

Keterangan:
$\bar{D}$ = rata-rata dari selisih nilai antara pasangan data
S_D = standar deviasi dari selisih nilai
n = jumlah pasangan data

2.6.2 Uji Signifikansi (Nilai P)

Uji signifikansi adalah prosedur statistik yang bertujuan untuk menentukan apakah hasil yang diperoleh dari suatu sampel mencerminkan efek yang nyata dalam populasi atau hanya terjadi secara kebetulan [19]. Nilai p adalah probabilitas untuk memperoleh nilai statistik uji yang sama atau lebih ekstrem dari nilai yang diamati, dengan asumsi bahwa hipotesis nol (H_0) benar [23].

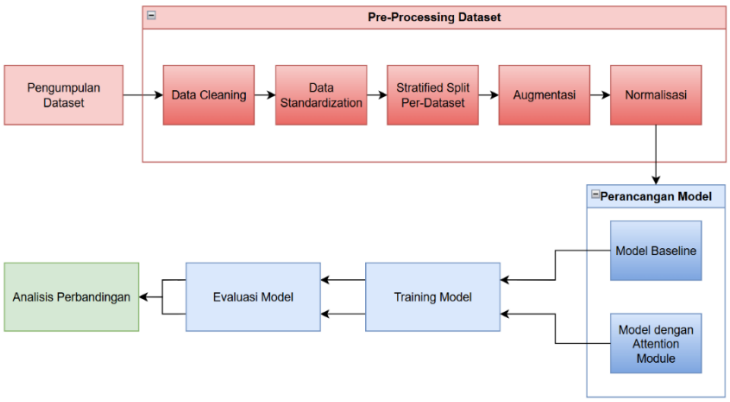
Kaidah pengambilan keputusan dalam uji signifikansi adalah sebagai berikut [24]:

- Jika nilai p < α, maka H_0 ditolak dan hasil dinyatakan **signifikan secara statistik**.
- Jika nilai p > α, maka H_0 tidak dapat ditolak dan hasil dinyatakan **tidak signifikan secara statistik**.

Tingkat signifikansi (alfa, α) adalah nilai ambang yang ditetapkan oleh peneliti sebelum melakukan uji statistik untuk menentukan apakah hasil dianggap signifikan secara statistik.

Dalam praktiknya, nilai p tidak dihitung secara manual, peneliti dapat menggunakan aplikasi statistik seperti SPSS, R, atau Microsoft Excel untuk menghitung nilai p.

3. METODE PENELITIAN



Gambar 1. Alur Penelitian

3.1 Metode Penelitian

3.1.1 Pengumpulan Dataset

Penelitian ini menggunakan tiga dataset citra penyakit kulit yang tersedia untuk publik secara gratis yaitu, HAM10000, DermNet, dan Mendeley Skin Disease Classification. Adapun dataset ini diunduh melalui website *Kaggle*.

Sampel pada tiap dataset tidak seimbang. Hal ini dapat menyebabkan model menjadi kurang akurat karena bias terhadap kelas mayoritas. Untuk mengatasi masalah ini, akan digunakan teknik *class weight* untuk memberikan bobot pada setiap kelas berdasarkan jumlah sampelnya

3.1.2 Preprocessing Dataset

Dataset yang telah dikumpulkan akan melalui serangkaian proses yang bertujuan untuk mempersiapkan data agar siap digunakan dalam proses pelatihan model.

Data cleaning

Dalam proses ini akan dilakukan langkah penanganan pada data yang tidak relevan, duplikat, atau data yang mengandung kesalahan.

Kanonisasi data

Pada tahap ini, akan dilakukan penetapan berbagai nama kelas yang merepresentasikan kelas yang sama menjadi satu nama yang sama. Proses ini dilakukan dengan menggabungkan atau memisahkan sampel dari kelas asal ke kelas dengan nama yang baru.

Penelitian ini akan menggunakan 28 kelas yang didapat dari menggabungkan seluruh kelas dari dataset yang digunakan yaitu *Acne, Actinic Keratoses, Alopecia, Atopic Dermatitis, Basal Cell Carcinoma, Benign Keratosis-like Lesions, Bullous, Cellulitis Impetigo, Dermatofibroma, Eczema, Exanthems, Herpes HPV, Light Diseases, Lupus, Melanocytic Nevi, Melanoma, Nail Fungus, Pigmentation Disorders, Poison Ivy, Psoriasis, Scabies Lyme, SJS-TEN, Systemic Disease, Tinea Candidiasis, Urticaria Hives, Vascular Lesions, Vasculitis,* dan *Wart Molluscum*.

Data split

Dataset kemudian akan di-*split* secara *stratified*. Hal ini dimaksudkan agar tidak ada data dari salah satu dataset mendominasi salah satu dari ketiga split nantinya. Dataset akan displit menjadi 3 bagian yaitu *train*, *validation*, dan *test* dengan rasio 80:10:10. Rasio split ini menggunakan penelitian [25] sebagai dasar penentuannya.

Augmentasi data

Augmentasi diterapkan hanya pada kelas dengan jumlah sampel yang kecil di dalam *split train*. Teknik augmentasi ini disebut *targeted data augmentation* [26]. Adapun augmentasi yang diberikan adalah sebagai berikut.

Tabel 1. Nilai augmentasi

Proses Augmentasi	Nilai
HorizontalFlipping	0.5
RandomRotation	20
ColorJitter	brigthness = 0.2, contrast = 0.4, saturation = 0.3

Setelah proses augmentasi, data train yang baru akan digunakan untuk menghitung *class weight* menggunakan fungsi *compute_class_weight* dari library *scikit-learn*.

Normalisasi

Pada penelitian ini, normalisasi dilakukan secara otomatis dengan menggunakan fungsi *v2.Normalize* dan *v2.ToImage* dari library *torchvision.transforms*.

3.1.3 Pembangunan Model MobileNetV3

Penelitian ini akan menggunakan 2 model utama yaitu MobileNetV3 dengan *custom head classifier* sebagai model *baseline* dan model *baseline* yang di-integrasi dengan CBAM pada arsitekturnya. Adapun posisi integrasi CBAM sebagai berikut:

1. CBAM setelah setiap *Inverted Residual Block* yang memiliki *Squeeze-and-Excitation Block*
2. CBAM tepat setelah *Squeeze-and-Excitation Block*
3. CBAM menggantikan *Squeeze-and-Excitation Block*

Keempat konfigurasi model akan menggunakan *custom head classifier* yang terdiri dari lapisan berikut:

1. *Fully connected layer* dengan 256 neuron
2. *Batch normalization*
3. Fungsi aktivasi *ReLU*
4. *Dropout* dengan rate 0.3 untuk mengurangi risiko *overfitting*
5. *Fully connected layer* dengan 28 neuron sebagai output

3.1.4 Training Model

Penelitian ini dilakukan dengan menggunakan pendekatan *transfer learning* dengan metode *feature extraction*. Artinya, model akan menggunakan *pre-trained weight* dan mengunci seluruh parameter *backbone* agar tidak ikut berlatih saat proses *training*.

3.1.5 Evaluasi Model

Evaluasi performa model dilakukan setelah model selesai dilatih. Metriks evaluasi yang digunakan mengacu pada metriks yang umum digunakan dalam penelitian klasifikasi seperti *accuracy*, *precision*, *recall*, *F1-Score,* dan *Confusion Matrix*. Perhitungan metriks-metriks tersebut dilakukan secara otomatis dengan fungsi *classification_report* dan *confusion_matrix* dari pustaka *sklearn.metrics*.

3.1.6 Analisis Model

Analisis performa dilakukan dengan membandingkan model baseline MobileNetV3 dan model usulan (dengan CBAM). Evaluasi mencakup nilai rata-rata (*mean*) untuk melihat performa rata-rata, serta standar deviasi untuk mengukur stabilitas model.

Untuk menguji signifikansi perbedaan performa, digunakan *paired t-test* dengan tingkat signifikansi $\alpha = 0{,}05$. Hipotesis nol (H_0) menyatakan tidak ada perbedaan signifikan, sedangkan hipotesis alternatif (H_1) menyatakan ada perbedaan signifikan. Jika *p-value* < 0,05, maka peningkatan yang diperoleh dianggap signifikan secara statistik. Perhitungan dilakukan menggunakan fungsi *ttest_rel* dari pustaka *scipy.stats*.

3.2 Desain Eksperimen

Penelitian ini menggunakan dua skenario utama: model baseline MobileNetV3 dan model usulan dengan integrasi CBAM pada backbone. Terdapat tiga konfigurasi peletakan CBAM:

1. CBAM A, CBAM setelah *Inverted Residual Block* yang memiliki *Squeeze-and-Excitation Block*
2. CBAM B, CBAM tepat setelah *Squeeze-and-Excitation Block*
3. CBAM menggantikan *Squeeze-and-Excitation Block*

Keempat model (baseline + tiga varian CBAM) dilatih dengan dataset yang sama dan *hyperparameter* identik untuk perbandingan yang adil.

Tabel 2. Hyperparameter pelatihan

Hyperparameter	Nilai
Optimizer	AdamW
Learning Rate	0.0001
L2 Regularization	0.0001
Batch Size	64
Maksimum Epoch	100
Early Stopping-monitor	Val_loss
Early Stopping-patience	5
Loss Function	Cross-entropy dengan *class weight*

Untuk mengurangi bias terhadap inisialisasi acak, setiap model dilatih dan dievaluasi 5 kali dengan *seed* tetap (42, 3407, 403, 2004, 2026).

4. PEMBAHASAN

4.1 Pengumpulan Data

Dataset diunduh dari situs web *Kaggle* dan diperoleh banyak data Dermnet berjumlah 19.559 dengan 23 kelas, HAM10000 berjumlah 10.015 dengan 7 kelas, dan Mendeley Skin Disease Classification (MSDC) berjumlah 9.548 dengan 5 kelas. Ketiga dataset memiliki jumlah sampel yang tidak sama, sehingga ketiga dataset mengalami kondisi *data imbalance*.

4.2 Preprocessing Data

4.2.1 Data Cleaning

Pada pembersihan data, dataset MSDC dikurangi dari 9.548 citra menjadi 341 citra asli dengan menghapus seluruh hasil augmentasi. Dataset HAM10000 disaring dari 10.015 citra menjadi 3.400 citra melalui penghapusan duplikasi berdasarkan *lesion_id* dan seleksi metode diagnosis

paling andal. Sementara itu, dataset DermNet yang semula berisi 19.500 citra dibersihkan secara manual dari citra buram, salah label, atau berisi objek non-kulit.

Tabel 3. Jumlah Pembersihan Data

Dataset	Data Original	Data Duplikat	Data Anomali	Data Akhir
DermNet	19.559	2.906	2.927	13.726
HAM10000	10.015	2.545	4.070	3.400
MSDC	9.548	9.207	-	341
TOTAL	**39.122**			**17.467**

4.2.2 Kanonisasi Data

Tahap kanonisasi data menghasilkan 28 kelas (telah dijelaskan pada poin 3.1.2). Walaupun telah diseragamkan, berdasarkan perhitungan sampel tiap kelas dari ketiga dataset, dataset tetap akan mengalami *data imbalance* ketika digabungkan nantinya.

4.2.3 Split Data

Dataset di-split secara stratifikasi dengan rasio 80:10:10, kemudian digabungkan menjadi 1 dataset besar berdasarkan splitnya.

Tabel 4. Proporsi Pembagian Dataset

Split / Dataset	Train	Validation	Test
Dermnet	10.981	1.373	1.372
HAM10000	2.719	340	341
MSDC	273	34	34
TOTAL	**13.973**	**1.747**	**1.747**

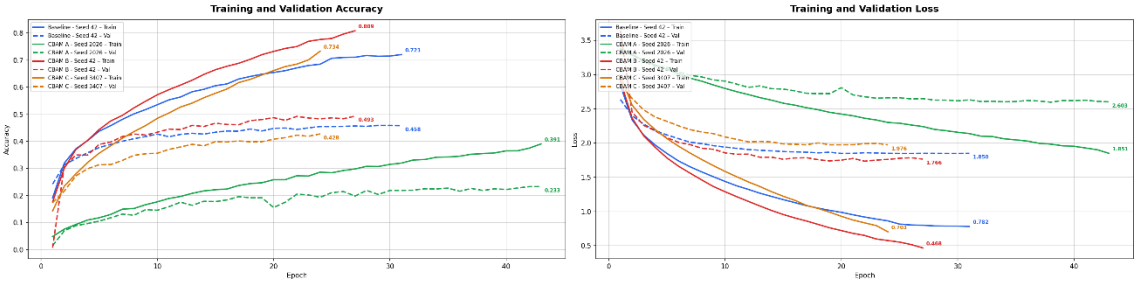
4.2.4 Augmentasi Data

Augmentasi data dilakukan pada kelas tertentu, kelas dengan sampel kecil seperti *alopecia*, *atopic dermatitis*, *dermatofibroma*, *nail fungus*, *pigmentation disorders*, *poison ivy*, *SJS-TEN*, dan *urticaria hives* di augmentasi. Berikut adalah

Kelas	Data Train	Hasil Augmentasi	Total Akhir
Alopecia	225	225	450
Atopic Dermatitis	182	182	364
Dermatofibroma	58	174	232
Nail Fungus	169	169	338
Pigmentation Disorders	78	234	312
Poison Ivy	212	212	424
SJS-TEN	90	270	360
Urticaria Hives	209	209	418

Proses augmentasi menambah sebanyak 1.675 data, menambah data *train* menjadi sebanyak 15.648 data.

4.3 Training Model



Gambar 2. Plot Akurasi dan Loss dari Training dan Validation

Gambar 2 adalah plot akurasi dan *loss* dari *training* dan *validation* yang diambil dari performa terbaik model pada *seed* tertentu. Model *baseline* dan CBAM B mendapatkan performa terbaik di *seed* 42, sedangkan CBAM A pada *seed* 2026, dan CBAM C pada *seed* 3407. Meski begitu, semua model mengalami *overfitting* yang ditandai dengan selisih antara nilai *training* dan *validation* yang cukup jauh.

4.4 Evaluasi Model

Tabel 5. Rangkuman Metrik Setiap Model

Metrik	**Model**			
	Baseline	**CBAM A**	**CBAM B**	**CBAM C**
Accuracy	0.4358 ± 0.0036	0.1631 ± 0.0317	0.4768 ± 0.0133	0.3959 ± 0.0233
Precision (Macro)	0.4360 ± 0.0108	0.1423 ± 0.0328	0.4635 ± 0.0168	0.3866 ± 0.0302
Recall (Macro)	0.4705 ± 0.0049	0.1959 ± 0.0405	0.4898 ± 0.0136	0.4314 ± 0.0190
F1-Score (Macro)	0.4441 ± 0.0066	0.1393 ± 0.0368	0.4721 ± 0.0156	0.3999 ± 0.0264

Berdasarkan tabel 5, model dengan konfigurasi CBAM B meraih performa terbaik dibandingkan model dengan CBAM konfigurasi lainnya. Model dengan konfigurasi CBAM A mengalami penurunan performa yang sangat drastis, sedangkan model dengan konfigurasi CBAM C tidak dapat menyamai performa model *baseline*.

4.5 Analisis Model

Berdasarkan hasil evaluasi, model *baseline* dan CBAM B selanjutnya diuji dengan menggunakan metode *paired t-test* untuk menunjukkan signifikansi dari perbedaan nilai yang diperoleh.

Tabel 6. Hasil kalkulasi *P value*

Metrik	**P Value**
Accuracy	0.0016
Precision (Macro)	0.0139
Recall (Macro)	0.0488
F1-Score (Macro)	0.0114

Seluruh nilai *p value* yang diperoleh (tabel 6) berada di bawah ambang batas signifikansi (0,05), dengan hasil tersebut, maka perbedaan performa (antara *baseline* dan CBAM B) dapat dikatakan signifikan dan hasil ini memberikan cukup bukti statistik untuk menolak H_0.

Perbedaan performa ketiga konfigurasi CBAM pada MobileNetV3 disebabkan oleh posisi integrasi dan interaksinya dengan arsitektur bawaan. Pada CBAM A, penempatan CBAM pada lapisan *Inverted Residual Block* di awal, dimana lapisan ini memiliki 40 channel, jika direduksi dengan rasio 16, maka akan menghasilkan bottleneck ekstrem (hanya 2 neuron MLP) sehingga

atensi kehilangan informasi penting dan performa menurun drastis. Pada CBAM C, CBAM menggantikan blok SE yang telah terlatih ImageNet, menyebabkan ketidakcocokan representasi dan gangguan pembelajaran, diperparah oleh data yang terbatas dan tidak seimbang. Sebaliknya, CBAM B menempatkan CBAM setelah blok SE tanpa menggantinya, sehingga terhindar dari bottleneck ekstrem dan tetap mempertahankan bobot terlatih, efeknya CBAM berhasil meningkatkan performa *baseline* secara signifikan, terutama pada kelas minoritas.

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, didapatkan hasil bahwa penerapan *Convolutional Block Attention Module* (CBAM) tepat setelah *SE Block* (konfigurasi CBAM B) arsitektur MobileNetV3 terbukti memberikan peningkatan performa secara signifikan (berdasarkan uji statistik) pada klasifikasi penyakit kulit pada rata-rata akurasi (0.041), *precision* (0.0275), *recall* (0.0193), dan *F1-score* (0.028). Namun dengan catatan bahwa letak integrasi CBAM sangat mempengaruhi hasil, hal ini dibuktikan dengan konfigurasi CBAM A (setelah *Inverted Residual Block* dengan *SE Block* di dalamnya) dan CBAM C (CBAM menggantikan *SE Block*) yang menghasilkan performa di bawah model baseline MobileNetV3 tanpa *attention module*.

6. DAFTAR PUSTAKA

[1] A. Irjayanti, A. Wambrauw, I. Wahyuni, and A. A. Maranden, "Personal Hygiene with the Incidence of Skin Diseases," *Jurnal Ilmiah Kesehatan Sandi Husada*, vol. 12, no. 1, pp. 169–175, Jun. 2023, doi: 10.35816/jiskh.v12i1.926.

[2] S. S. Mohammed and J. M. Al-Tuwaijari, "Skin Disease Classification System Based on Machine Learning Technique: A Survey," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1076, no. 1, p. 012045, Feb. 2021, doi: 10.1088/1757-899x/1076/1/012045.

[3] Z. Ullah, M. Hong, T. Mahmood, and J. Kim, "Systematic Integration of Attention Modules into CNNs for Accurate and Generalizable Medical Image Classification," *Mathematics*, vol. 13, no. 22, Nov. 2025, doi: 10.3390/math13223728.

[4] A. Aboulmira, H. Hrimech, and M. Lachgar, "Comparative Study of Multiple CNN Models for Classification of 23 Skin Diseases," *International journal of online and biomedical engineering*, vol. 18, no. 11, pp. 127–142, 2022, doi: 10.3991/ijoe.v18i11.32517.

[5] S. A. Hanum, A. Dey, and M. A. Kabir, "An Attention-Guided Deep Learning Approach for Classifying 39 Skin Lesion Types," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.05991

[6] Hartatik, A. Aziz, R. Hartono, and R. Abdilah, "Decision Support System for Detection of Skin Diseases in Smart Health development planning," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1757-899X/858/1/012051.

[7] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 679–688. doi: 10.1016/j.procs.2018.05.069.

[8] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861

[9] A. Howard *et al.*, "Searching for MobileNetV3," Nov. 2019, [Online]. Available: http://arxiv.org/abs/1905.02244

[10] J. Wang *et al.*, "Integrating Biological and Machine Intelligence: Attention Mechanisms in Brain-Computer Interfaces," Jul. 2025, [Online]. Available: http://arxiv.org/abs/2502.19281

[11] Z. Zhang and M. Wang, "Convolutional Neural Network with Convolutional Block Attention Module for Finger Vein Recognition," Feb. 2022, [Online]. Available: http://arxiv.org/abs/2202.06673

[12] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module."
[13] M.-H. Guo *et al.*, "Attention Mechanisms in Computer Vision: A Survey," Nov. 2021, doi: 10.1007/s41095-022-0271-y.
[14] S. Sathyanarayanan and B. R. Tantri, "Confusion Matrix-Based Performance Evaluation Metrics," *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024, doi: 10.53555/ajbr.v27i4s.4345.
[15] J. C. Obi, "A comparative study of several classification metrics and their performances on data," *World Journal of Advanced Engineering Technology and Sciences*, vol. 8, no. 1, pp. 308–314, Feb. 2023, doi: 10.30574/wjaets.2023.8.1.0054.
[16] S. Beddar-Wiesing, A. Moallemy-Oureh, M. Kempkes, and J. M. Thomas, "Absolute Evaluation Measures for Machine Learning: A Survey," Jul. 2025, [Online]. Available: http://arxiv.org/abs/2507.03392
[17] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, no. 1, Mar. 2024, doi: 10.1038/s41598-024-56706-x.
[18] M. C. Hinojosa Lee, J. Braet, and J. Springael, "Performance Metrics for Multilabel Emotion Classification: Comparing Micro, Macro, and Weighted F1-Scores," *Applied Sciences (Switzerland)*, vol. 14, no. 21, Nov. 2024, doi: 10.3390/app14219863.
[19] C. Andrade, "The P value and statistical significance: Misunderstandings, explanations, challenges, and alternatives," May 01, 2019, *Wolters Kluwer Medknow Publications*. doi: 10.4103/IJPSYM.IJPSYM_193_19.
[20] R. Akbar, U. S. Sukmawati, and K. Katsirin, "Analisis Data Penelitian Kuantitatif," *Jurnal Pelita Nusantara*, vol. 1, no. 3, pp. 430–448, Jan. 2024, doi: 10.59996/jurnalpelitanusantara.v1i3.350.
[21] A. Dhianti Putri, R. Sayyida Hilmia, S. Almaliyah, S. Permana, and P. Studi Bimbingan dan Konseling, "PENGAPLIKASIAN UJI T DALAM PENELITIAN EKSPERIMEN," *Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika*, vol. 4, no. 3, pp. 1978–1987, Dec. 2023, doi: 10.46306/lb.v4i3.
[22] A. I. Talikan *et al.*, "ON PAIRED SAMPLES T-TEST: APPLICATIONS, EXAMPLES AND LIMITATIONS," *Ignatian International Journal for Multidisciplinary Research*, pp. 943–951, Apr. 2024, doi: 10.5281/zenodo.10987546.
[23] Nuryadi, T. D. Astuti, E. S. Utami, and M. Budiantara, *DASAR-DASAR STATISTIK PENELITIAN*. Yogyakarta: SIBUKU MEDIA, 2017. [Online]. Available: www.sibuku.com
[24] S. Kwak, "Are Only p-Values Less Than 0.05 Significant? A p-Value Greater Than 0.05 Is Also Significant!," *J. Lipid Atheroscler.*, vol. 12, no. 2, pp. 89–95, May 2023, doi: 10.12997/jla.2023.12.2.89.
[25] H. Bichri, A. Chergui, and M. Hain, "Investigating the Impact of Train / Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets," 2024. [Online]. Available: www.ijacsa.thesai.org
[26] A. Mikołajczyk-Bareła, M. Ferlin, and M. Grochowski, "Targeted data augmentation for improving model robustness against systematic bias," Mar. 11, 2024. doi: 10.21203/rs.3.rs-3953661/v1.