

## **Dendritic ShuffleNetV2 Model for Alzheimer's Disease Imaging Classification**

**Riandika Fathur Rochim<sup>1\*</sup>, I Gusti Ngurah Lanang Wijayakusuma<sup>2\*</sup>**

\* Matematika, Universitas Udayana

[riandikafathurrochim@gmail.com](mailto:riandikafathurrochim@gmail.com)<sup>1</sup>, [lanang\\_wijaya@unud.ac.id](mailto:lanang_wijaya@unud.ac.id)<sup>2</sup>,

---

### **Article Info**

#### **Article history:**

Received 2025-06-03

Revised 2025-06-27

Accepted 2025-07-03

---

#### **Keyword:**

*Alzheimer's Classification,  
Deep Learning,  
ShuffleNet,  
Dendritic Neural Model.*

---

### **ABSTRACT**

This study investigates the integration of a dendritic neural model (DNM) into the ShuffleNetV2 architecture to enhance Alzheimer's stage classification from MRI scans. The proposed “Dendritic ShuffleNetV2” retains the original network’s computational cost (0.31 GFLOPs) while incurring only a 1.6% increase in parameter count (from 2.48 M to 2.52 M) and achieves faster convergence (15 epochs versus 22 epochs). Experiments were conducted on a four-class Alzheimer’s MRI dataset comprising Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented categories. Compared to the baseline ShuffleNetV2, the Dendritic variant yielded an average accuracy improvement of 0.79%, with corresponding gains of approximately 0.8% in weighted precision, recall, and F1-score. Confusion matrix analysis revealed persistent overlap between the Very Mild and Mild Demented classes, although overall discrimination—particularly for the majority and early-stage classes—remained robust. Training stability was maintained without significant overfitting.



This is an open access article under the [CC-BY-SA](#) license.

---

### **I. PENDAHULUAN**

Penyakit Alzheimer atau *Alzheimer's Disease* (AD) adalah gangguan neurodegeneratif progresif yang menurunkan fungsi kognitif, memori, dan perilaku pasien secara masif [1]. Kondisi ini ditandai oleh penumpukan plak  $\beta$ -amiloid dan tau di otak [1], [2]. Karena sifatnya yang progresif dan saat ini studi epidemiologi menunjukkan bahwa terdapat penurunan prevalensi AD melalui pengobatan jangka panjang menggunakan obat antiinflamasi nonsteroid (oAINS) [3], diagnosis dini AD sangat penting untuk deteksi sedini mungkin guna memperlambat perkembangan penyakit dan meningkatkan kualitas hidup pasien [2], [4]. Deteksi awal ini biasanya dilakukan melalui analisis pencitraan otak (seperti MRI atau PET) dan penilaian kognitif [1].

Baru-baru ini, pendekatan *deep learning* terutama *convolution neural network* (CNN) pada citra medis semakin diandalkan untuk mengotomasi klasifikasi AD dari gambar otak [1]. Seperti penelitian dari Austin et al. yang dipublikasi pada tahun 2024 melakukan pendekatan *deep learning* untuk klasifikasi AD menggunakan model ConvNeXt dengan hasil akurasi uji 75% untuk klasifikasi empat tingkat keparahan Alzheimer [5]. Dan penelitian dari Nithya et al. pada tahun

2023 juga melakukan klasifikasi AD menggunakan model ResNet-50 dengan hasil akurasi 95% untuk klasifikasi AD vs normal [6]. Namun, tantangan utama muncul saat mengembangkan model yang tidak hanya akurat tetapi juga mampu diterapkan pada perangkat komputasi terbatas (*edge device*) seperti perangkat seluler atau portabel, yang memiliki keterbatasan memori dan pemrosesan.

Dalam konteks komputasi *edge*, dibutuhkan model *deep learning* yang ringan dan efisien. Model konvensional berlapis dalam seperti VGG, ResNet atau ConvNeXt umumnya memiliki puluhan jutaan parameter dan membutuhkan daya komputasi besar, sehingga sulit dijalankan di perangkat portable [7]. Oleh karena itu, arsitektur CNN ringan (*lightweight*) seperti MobileNet dan ShuffleNet dikembangkan khusus untuk mempercepat inferensi dan mengurangi konsumsi sumber daya tanpa mengorbankan akurasi secara drastis [7]. Sebagai contoh, penelitian pada tahun 2023 menggunakan ShuffleNetV1 untuk ekstraksi fitur citra MRI Alzheimer berhasil mendapatkan akurasi tinggi (95% untuk klasifikasi AD vs normal) dengan hanya 3,42 juta parameter dan 6,08 GFLOPs [7]. Desain model tersebut terbukti efisien secara komputasi sehingga dapat memproses data dalam jumlah besar secara

cepat dan akurat, sesuai kebutuhan perangkat yang terbatas daya komputasinya [7].

ShuffleNetV2 merupakan contoh arsitektur CNN ringan yang dirancang dengan prinsip efisiensi tersebut. ShuffleNetV2 diperkenalkan dengan fokus pada metrik kecepatan dan biaya akses memori rendah, serta menerapkan aturan praktis untuk desain jaringan yang efisien [8]. Arsitektur ini menggunakan operasi *channel split*, *depthwise convolution*, dan *channel shuffle* untuk membagi dan menggabungkan jalur fitur secara efektif, sehingga meringankan beban komputasi dan memungkinkan eksekusi yang lebih cepat pada perangkat *edge* [8]. Dibandingkan ShuffleNetV1 dan MobileNetV2, ShuffleNetV2 menunjukkan peningkatan akurasi 3–4% pada batas FLOPs yang sama, serta *throughput* lebih tinggi pada CPU/GPU [8].

*Dendritic neural model* (DNM) merupakan pendekatan yang terinspirasi oleh sifat biologis neuron, khususnya struktur dendritiknya. Berbeda dengan neuron McCulloch-Pitts standar yang linear, DNM menampilkan struktur pohon dendritik dan fungsi aktivasi yang unik [9]. DNM terinspirasi dari temuan neurofisiologi bahwa cabang dendrit menghasilkan lonjakan lokal (*Calcium spikes*) dan berfungsi seperti *micro-perceptrons*, mereka dapat melakukan operasi logika AND/OR dan *gating* sebelum sinyal dikumpulkan di badan sel serta telah ditunjukkan meningkatkan efisiensi dan kekuatan *artificial neural network* (ANN) modern [10]. Struktur ini memproses informasi secara non-linear yang kompleks, secara detail sinyal input diproses melalui lapisan sinaptik dan dendritik yang saling mengalikan atau menambahkan (mirip operasi logika), sebelum diakumulasi pada lapisan membran dan soma [9], [11]. Karakteristik ini membuat DNM mampu melakukan operasi logika yang lebih kompleks dan aproksimasi fungsi secara lebih akurat dibandingkan model neuron konvensional [9]. Selain itu, DNM relatif mudah diimplementasikan dan memiliki sifat interpretabilitas yang baik berkat struktur biologis yang jelas [9]. Dalam konteks klasifikasi citra visual, motivasi penggunaan DNM adalah potensi peningkatan performa klasifikasi lewat pemanfaatan kemampuan pemrosesan non-linear *dendritic* [12], [13]. Beberapa penelitian menyarankan bahwa DNM yang dipelajari dengan baik dapat mengungguli model neuron tradisional dan dapat menjadi blok bangunan untuk generasi berikutnya dari algoritma pembelajaran dalam [14].

Penelitian ini bertujuan melakukan perbandingan komparatif antara ShuffleNetV2 dan *dendritic* ShuffleNetV2 dimana merupakan integrasi DNM ke dalam ShuffleNetV2 untuk tugas klasifikasi penyakit Alzheimer berbasis citra otak (MRI). Fokus utamanya adalah menguji efisiensi kedua model tersebut pada perangkat dengan sumber daya terbatas (misalnya *edge device*), sekaligus mengeksplorasi apakah struktur DNM dapat memberikan keuntungan performa dalam klasifikasi AD. Dengan melakukan penelitian ini, diharapkan akan terungkap *trade-off* antara kompleksitas arsitektur dan akurasi klasifikasi, serta potensi penggunaan DNM sebagai alternatif inovatif dalam deteksi dini AD. Hasil

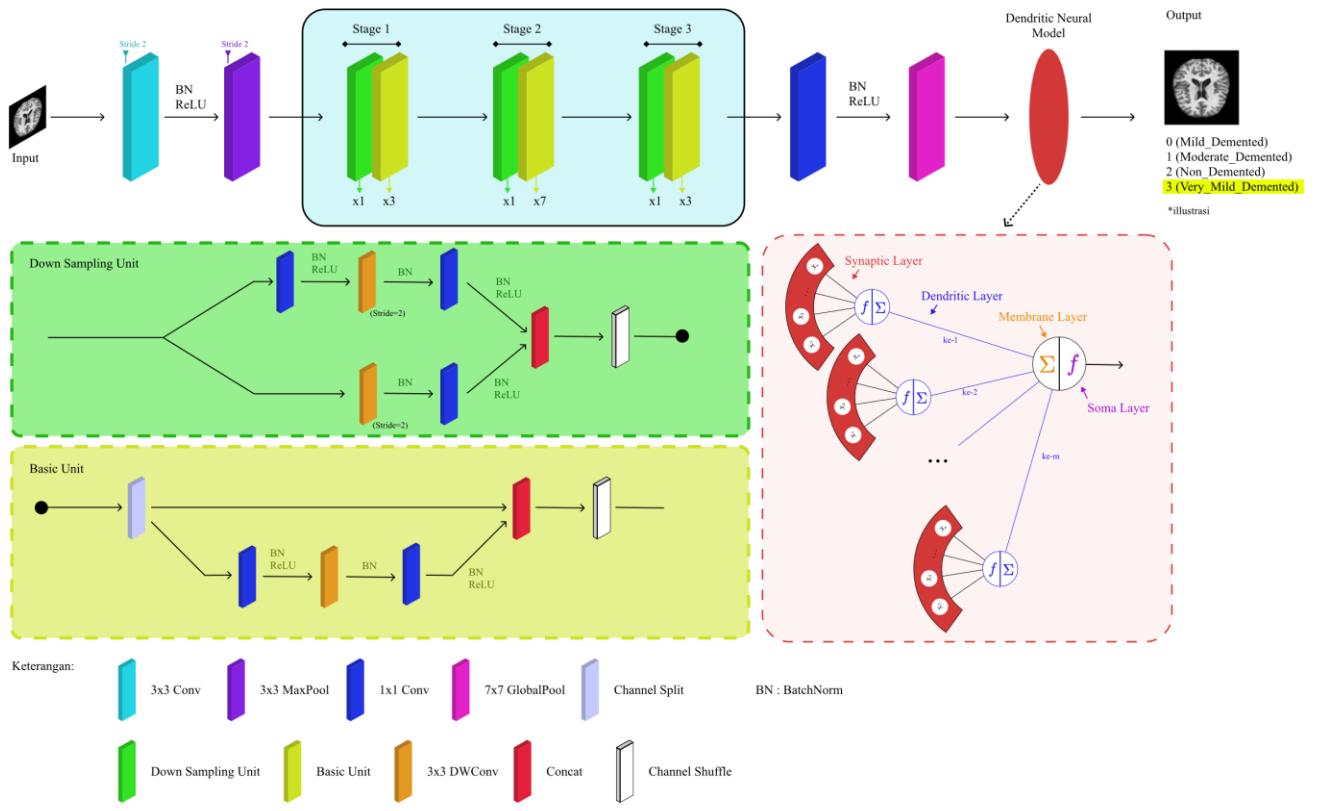
dan analisis dari perbandingan ini diharapkan dapat memberikan wawasan penting bagi pengembangan model deep learning efisien untuk aplikasi medis yang dapat berjalan langsung di perangkat terukur (*mobile/edge*).

Pendekatan di atas didukung oleh beberapa literatur akademik. Penelitian Liu et al. dan Wang et al. yang dipublikasikan pada tahun 2025 mengintegrasikan DNM ke dalam CNN (ShuffleNetV2 dan ResNeXt) untuk meningkatkan akurasi klasifikasi citra medis [11], [15]. Penelitian Ding et al. yang dipublikasikan pada tahun 2023 menunjukkan bahwa lapisan DNM multi-neuron (*Multi-Layer Dendritic Net*) dapat mengantikan perceptron di akhir CNN untuk klasifikasi digit MNIST [16]. Konsep *multi-out* juga telah dipelajari untuk memperluas DNM ke klasifikasi multi-kelas [16]. Lapisan DNM dapat menjadi alternatif untuk mengganti *fully connected layer* terakhir CNN dalam tugas klasifikasi multi-kelas, dengan asumsi keluaran skalar per neuron digabungkan menjadi vektor kelas dan diberi aktivasi *softmax* [11], [15], [17].

## II. METODE

Untuk memudahkan dan generalisasi penjelasan model DNM yang akan diintegrasikan ke model ShuffleNetV2, akan diilustrasikan pada Gambar 1.

Arsitektur ShuffleNetV2 terdiri dari beberapa komponen, seperti konvolusi  $3 \times 3$  ( $3 \times 3$  conv), *max pooling*, tiga *stage*, konvolusi  $1 \times 1$  ( $1 \times 1$  conv), dan *global pooling* serta *fully connected layer* sebagai lapisan akhirnya. Dalam tiga *stage* tersebut terdiri dari dua bagian dengan frekuensi pengulangan bervariasi di setiap *stage*, yaitu *basic unit* dan *down sampling unit*. Dan secara keseluruhan, arsitektur ini dibangun dari *basic unit* dan *down sampling unit*. Di bagian *basic unit*, fitur *input* dibagi menjadi dua lewat *channel split* dimana masing-masing panjang kanalnya sama dengan setengah dari jumlah kanal awal yang kemudian diteruskan ke cabang kanan yang berisikan konvolusi  $1 \times 1$  ( $1 \times 1$  conv) dan konvolusi *depthwise*  $3 \times 3$  ( $3 \times 3$  dwconv) dan cabang kiri dengan berjalan secara paralel. Hasil dari operasi dari cabang kanan dan kiri akan digabung (*concat*) dan dilakukan *channel shuffle* yang berguna untuk bertukar informasi antar kanal. Pada *down-sampling unit*, fitur *input* langsung dibagi menjadi dua cabang, dimana cabang kanan menjalani dua operasi berurutan yaitu konvolusi  $1 \times 1$  dan *depthwise*  $3 \times 3$  dengan *stride* 2 dan cabang kiri menjalani *depthwise*  $3 \times 3$  dengan *stride* 2, diikuti dengan konvolusi  $1 \times 1$ . Setelah operasi penggabungan (*concat*), jumlah kanal output menjadi dua kali jumlah kanal awal dan kemudian dilakukan *channel shuffle*. Berbeda dengan *basic unit*, *down-sampling unit* tidak menggunakan *channel split*, karena tujuannya untuk meningkatkan bobot komputasi dengan memperluas jumlah kanal dan meningkatkan ruang ekstraksi fitur secara langsung. Setelah itu, fitur citra output akan memiliki jumlah kanal dua kali lipat dari fitur citra asli. Kemudian, hasil fitur citra akan menjalani konvolusi  $1 \times 1$  dan *global pooling*.



Gambar 1. Arsitektur integrasi DNM dengan ShuffleNetV2.

Arsitektur model *dendritic* ShuffleNetV2 dibangun berdasarkan arsitektur dasar ShuffleNetV2. Perbedaan utamanya terletak pada lapisan akhir model untuk klasifikasi, dimana ShuffleNetV2 menggunakan *fully connected layer* sedangkan *dendritic* ShuffleNetV2 menggunakan lapisan DNM. Dalam klasifikasi, DNM diharapkan menunjukkan efisiensi dan stabilitas dalam klasifikasi jenis label AD. Secara keseluruhan, struktur DNM secara konsisten terdiri dari empat modul lapisan, yaitu lapisan *synaptic*, lapisan *dendritic*, lapisan *membrane*, dan lapisan *soma*. Lapisan *synaptic* dalam *neural network* berperan sebagai penghubung neuron-neuron dari satu dendrit ke sel neuron yang berbeda-beda, mengirim informasi secara umpan balik dari *presynaptic neuron* ke *postsynaptic neuron*. Dalam *synaptic layer*, menggunakan fungsi aktivasi untuk menghubungkan masukan ke- $i$  ( $i = 1, 2, \dots, N$ ) menuju ke dendrit ke- $j$  ( $j = 1, 2, \dots, M$ ), yang didefinisikan sebagai berikut:

$$Y_{ij} = \text{ReLU}(k \cdot (w_{ij}x_i - q_{ij}))$$

dimana  $N$  dan  $M$  masing-masing merupakan banyaknya masukan dan dendrit,  $\text{ReLU}()$  merupakan fungsi aktivasi yang akan diterapkan pada  $f$ ,  $k$  merupakan konstanta positif dengan nilai yang digunakan 0.5 [11],  $w_{ij}$  dan  $q_{ij}$  merupakan parameter pelatihan yang dibuat secara acak pada interval  $[0, 1]$ .

Hasil tersebut kemudian dikirim dari lapisan *synaptic* ke lapisan *dendritic*, yang dimana menerapkan operasi penjumlahan sebagai representasi operator logika OR untuk mencegah *vanishing problems* apabila menggunakan operator logika AND dengan representasi operasi perkalian [15]. Fungsi hasil untuk cabang *dendritic* ke- $j$  dinyatakan sebagai berikut:

$$Z_j = \sum_{i=1}^N Y_{ij}$$

dimana  $N$  merupakan banyaknya masukan.

Lapisan *membrane* mengumpulkan informasi yang dikirimkan dari lapisan *dendritic* dan melakukan penjumlahan atas informasi tersebut sebagai berikut:

$$V = \sum_{i=1}^M Z_j$$

dimana  $M$  merupakan banyaknya dendrit.

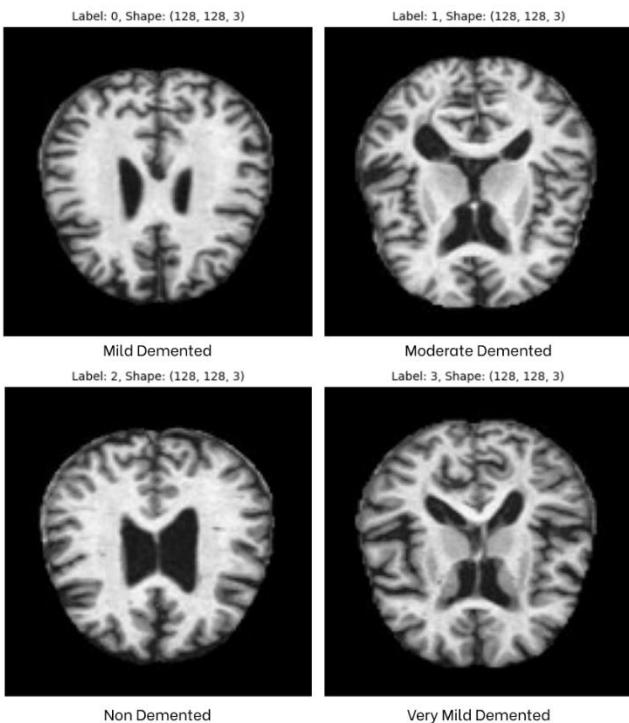
Setelah pemrosesan lapisan demi lapisan selesai, lapisan *soma* menerima sinyal dari lapisan *membrane*, dan hasil akhirnya ditentukan menggunakan fungsi  $\text{ReLU}()$ , yang dibentuk sebagai berikut:

$$O = \text{ReLU}(k_s \cdot (V - q_s))$$

dengan  $k_s$  dan  $q_s$  merupakan parameter pelatihan yang diinisialisasi secara acak pada interval  $[0,1]$ , lapisan DNM kemudian diintegrasikan ke dalam arsitektur ShuffleNetV2 dengan menggantikan lapisan *fully-connected* pada bagian ekor model, dengan fitur  $1 \times 1 \times C$  hasil dari *global pooling* dimasukan ke lapisan DNM dimana  $C$  merupakan jumlah *channel* yang dihasilkan dari *global pooling*[11] Dalam konfigurasi ini, *backbone* ShuffleNetV2 tetap utuh, dan lapisan DNM ditambahkan sebagai *classifier* baru yang dilatih secara *end-to-end* untuk digunakan dalam klasifikasi AD.

#### A. Persiapan Dataset

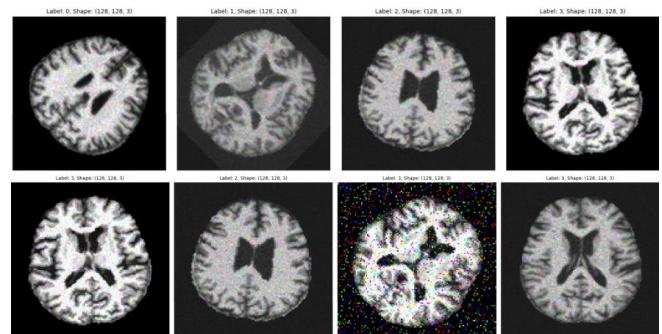
Dataset yang digunakan dalam penelitian ini adalah “*Alzheimer MRI (4 class of image)*” dari platform Hugging Face. Dataset ini diunduh melalui laman [Obscure-Entropy/Alzheimer-MRI · Datasets at Hugging Face](#), dimana dataset ini bersifat terbuka dan berlisensi MIT. Dataset ini berisikan gambar dari scan MRI otak manusia yang memiliki empat kategori kondisi, yaitu *Mild Demented*, *Moderate Demented*, *Non Demented*, dan *Very Mild Demented*. Dataset ini telah tersedia pembagian data secara langsung dimana terdapat data pelatihan (*train*), data uji (*test*), dan data validasi (*validation*), serta data augmentasi dari hasil *preprocessing* data pelatihan (*augmentation*) yang telah ditambahkan *noise*, rotasi, *flip-horizontal*, *flip-vertical*, *brightness*, ataupun perubahan kontras warna. Masing-masing jumlah pembagian data adalah 4.096 (64%), 1.024 (16%), 1.280 (20%), dan 4.096. Sehingga total keseluruhan data berjumlah 10.496 data dan data aslinya berjumlah 6.400 data.



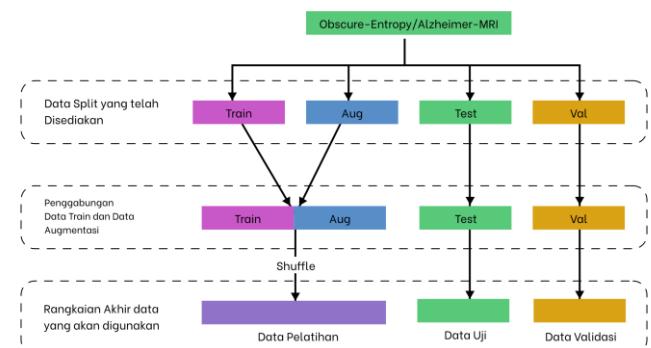
Gambar 2. Dataset Alzheimer MRI Disease di masing-masing kategori

Dataset juga akan dilakukan *preprocessing* data untuk meningkatkan kualitas data dan kinerja model yang diatur sebagai berikut.

- Data pada awal mulanya dibagi menjadi tiga bagian: *train*, *test* dan *validation*
- Data *train* telah dilakukan augmentasi dengan diberikan *noise*, rotasi, *flip-horizontal*, *flip-vertical*, *brightness*, ataupun perubahan kontras warna, sehingga terdapat bagian data baru, bernama data *augmentation*.
- Dikarenakan data *augmentation* berformat RGB, maka akan diubah format gambar pada data *train* yang sebelumnya *grayscale* menjadi format RGB.
- Ukur ulang gambar menjadi 224x224 untuk optimalkan performa arsitektur model [8].
- Melakukan normalisasi gambar sesuai dengan statistik bobot ImageNet
- Menggabungkan data pelatihan dan data augmentasi yang sebelumnya disediakan menjadi data pelatihan yang baru dengan bertujuan untuk memperkaya variasi data.
- Data pelatihan yang baru akan diacak posisinya secara *random* untuk mengoptimalkan pelatihan model (*shuffle*). Bisa dilihat pada Gambar 4.



Gambar 3. Hasil augmentasi secara acak yang telah disediakan



Gambar 4. Rancangan membuat rangkaian dataset akhir yang akan digunakan nantinya

Adapun informasi sebaran jumlah gambar pada *dataset* terhadap empat kelas baik secara jumlah proporsi keseluruhan tanpa hasil augmentasi maupun di masing-masing pembagian data akan ditunjukkan dalam bentuk tabel sebagai berikut.

TABEL 1

INFORMASI PROPORSI KELAS DATASET MELIPUTI DATA PELATIHAN AWAL, DATA PENGUJIAN, DAN DATA VALIDASI

Label	Jumlah Proporsi
Mild Demented	796
Moderate Demented	64
Non Demented	3200
Very Mild Demented	2240

TABEL 2

INFORMASI DATASET

Label	Data Train	Data Test	Data Valid	Data Train setelah Augmentasi
Mild Demented	579	172	145	1158
Moderate Demented	43	15	6	86
Non Demented	2044	634	522	4088
Very Mild Demented	1430	459	351	2860

### B. Tata Laksana Percobaan dan Evaluasi Model

Semua percobaan dilakukan menggunakan Python 3.10 dan Pytorch versi 1.13 di platform Google Colab dengan dijalankan di perangkat T4 GPU yang telah disediakan pada platform tersebut. Model yang akan digunakan dalam penelitian ini adalah ShuffleNetV2 dan MobileNetV2. Semua model dalam penelitian ini dilatih menggunakan Adam optimizer dengan learning rate 0.001. Selama pelatihan, ukuran batch diatur sebesar 32 dan learning rate dikurangi setengahnya setiap 5 epoch. Jumlah cabang pada DNM diatur sebanyak 8 untuk memaksimalkan performa klasifikasi pada dataset citra medis. Melalui proses eksperimen, nilai loss dari berbagai model pada setiap dataset terlihat mulai stabil setelah 30 epoch pelatihan. Oleh karena itu, baik model pembanding maupun dengan DNM dilatih selama total 30 epoch. Untuk mencegah model overfitting dalam pelatihan, akan digunakan mekanisme early stopping dengan patience selama lima epoch, yang secara otomatis menghentikan pelatihan apabila tidak terjadi peningkatan pada nilai validation loss dalam lima epoch berturut-turut. Strategi ini tidak hanya mempercepat waktu pelatihan, tetapi juga menjaga generalisasi model terhadap data yang tidak terlihat selama pelatihan. Dengan demikian, evaluasi yang dilakukan pada penelitian model ini menggunakan epoch optimal.

Untuk mengevaluasi kinerja model secara akurat, penelitian ini menggunakan empat metrik kinerja klasifikasi: akurasi (Acc), precision (Pre), recall (Rec), dan F1-Score (F1) dengan berdasarkan weighted average. Serta menggunakan dua metrik efisiensi seperti kecepatan inferensi (FLOPs atau FL) dan total parameter yang digunakan (Par) yang merepresentasikan total memori yang dibutuhkan.

### C. Penanganan Kelas Tak Seimbang

Masalah kelas tak seimbang atau *class imbalanced* sangat penting diatasi untuk mencegah model menjadi bias ke kelas mayoritas (*Non Demented*) dan pengabaian kelas minoritas (*Moderate Demented*). Dalam penelitian ini, kami memberikan pembobotan pada masing-masing kelas atau label data pelatihan (*train*). Dalam percobaan kami, terjadi overfitting dimana model terlalu agresif untuk klasifikasi kelas minoritas, sehingga diterapkan batas nilai yang dinyatakan sebagai berikut.

$$w_c = \min\left(\sqrt{\frac{N_{total}}{N_c}}, 4.5\right)$$

dimana  $w_c$  merupakan bobot pada kelas- $c$ ,  $N_{total}$  merupakan total data,  $N_c$  adalah total data pada kelas- $c$ , dan min() merupakan fungsi minimum. Hasil bobot tersebut kemudian akan dinormalisasikan dengan sebagai berikut.

$$\widehat{w}_c = \frac{w_c}{\sum_{i=1}^c w_i}$$

dimana  $\widehat{w}_c$  merupakan hasil normalisasi bobot pada kelas- $c$  dan  $w_c$  merupakan bobot pada kelas- $c$  yang kemudian dibagi dengan total seluruh bobot yang ada. Bobot di setiap kelas nantinya akan diterapkan pada *criterion* dengan *cross entropy loss* untuk fungsi loss pada model.

### D. Konfigurasi Hyperparameter M

Untuk analisis secara lanjut, dilakukan hyperparameter pada jumlah dendrit yang akan digunakan. Dengan bertujuan untuk mengoptimalkan performa model sehingga kemampuan klasifikasi model akan optimal apabila diberikan tugas klasifikasi AD. Untuk menyelidiki pengaruh parameter jumlah dendrit (M) dalam DNM terhadap model, dilakukan eksperimen pada dataset Alzheimer MRI. Hasil untuk berbagai nilai parameter M dicatat dalam Tabel 3.

TABEL 3  
HASIL HYPERPARAMETER M (WEIGHTED AVERAGE)

M	Acc (%)	Pre (%)	Rec (%)	F1 (%)
6	95,47	95,57	95,47	95,44
8	<b>96,41</b>	<b>96,41</b>	<b>96,41</b>	<b>96,41</b>
10	96,33	96,37	96,33	96,32
12	95,94	95,96	95,94	95,94
14	96,25	96,29	96,25	96,24
16	94,69	94,69	94,69	94,67

Berdasarkan hasil penelitian bahwa jika jumlah dendrit (M) terlalu sedikit, model tidak dapat sepenuhnya menangkap informasi fitur dari data. Namun, jika jumlah dendrit (M) terlalu banyak, dapat menyebabkan overfitting. Ketika  $M = 8$ , model menunjukkan performa yang secara signifikan lebih baik di setiap metrik evaluasi dan mencapai kinerja klasifikasi optimal.

### III. HASIL DAN PEMBAHASAN

Pada Tabel 4, hasil komparasi dengan model yang digunakan dalam penelitian ini memperlihatkan bahwa model ShuffleNetV2 yang diintegrasikan dengan DNM (Dendritik ShuffleNetV2) menunjukkan performa yang lebih baik dalam efisiensi pelatihan dibandingkan dengan tanpa menggunakan DNM. Namun, jumlah parameter yang digunakan lebih tinggi dibandingkan menggunakan ShuffleNetV2 biasa.

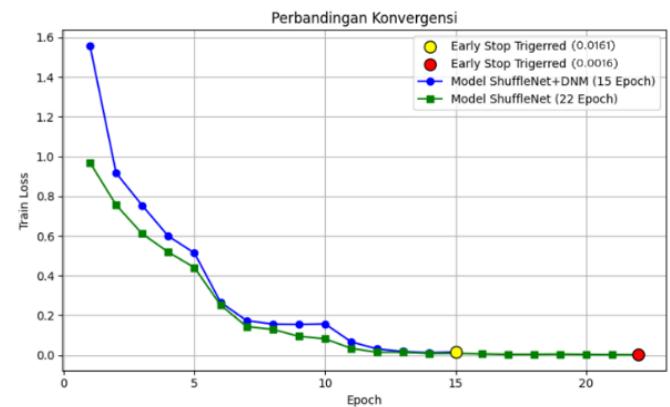
TABEL 4  
PERBEDAAN HASIL EVALUASI ANTAR MODEL (WEIGTHED AVERAGE)

Model	FL (G)	Par (M)	Acc (%)	Pre (%)	Rec (%)	F1 (%)
ShuffleNetV2	<b>0,31</b>	2,48	95,62	95,6	95,6	95,3
Dendritik ShuffleNetV2	<b>0,31</b>	2,52	<b>96,41</b>	<b>96,4</b>	<b>96,4</b>	<b>96,4</b>
MobileNetV2	0,64	<b>2,23</b>	<b>96,8</b>	<b>96,9</b>	<b>96,8</b>	<b>95,8</b>

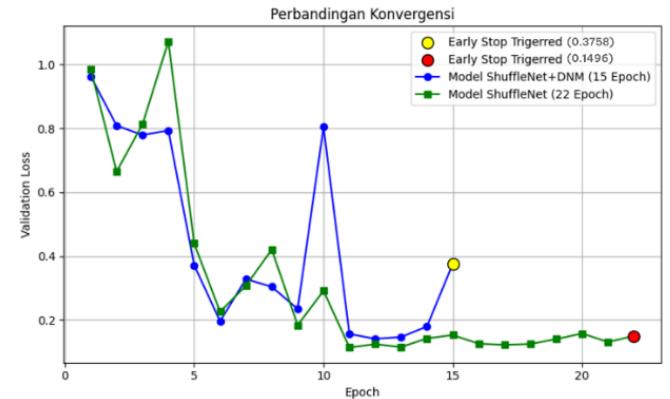
Walaupun menggunakan DNM memiliki sekitar 1,61% lebih banyak penggunaan parameter dibandingkan ShuffleNetV2 yang standar, peningkatan tersebut diimbangi dengan peningkatan akurasi, presisi, recall, dan F1-score, sehingga dapat dianggap sebagai *trade-off* yang layak dalam konteks peningkatan performa model. Serta peningkatan ini membantu pada perbaikan kinerja klasifikasi secara keseluruhan, sehingga tetap efisien dan masih bisa dibenarkan dari sisi kompleksitas model. Dengan demikian, penambahan DNM pada ShuffleNetV2 menjadikan model yang efisien tinggi dibandingkan dengan MobileNetV2 tanpa menurunkan hasil evaluasi dari *baseline* ShuffleNetV2 sebelumnya, justru meningkatkan akurasi tanpa mengorbankan efisiensi dengan ditandai GFLOPs yang sama (0,31 GFLOPs < 0,64 GFLOPs).

Hasil pelatihan antara juga memiliki grafik perbandingan konvergensi di *training loss* dan *validation loss*. Gambar 5 menunjukkan bahwa model ShuffleNetV2 standar membutuhkan 22 epoch untuk pelatihan yang optimal dengan *train loss* akhir 0.0016 dan akurasi pelatihan sebesar 99,98%. Sementara saat menggunakan DNM, model membutuhkan 15 epoch untuk pelatihan yang optimal dengan *train loss* akhir 0.0161 dan akurasi pelatihan sebesar 99,51%. Meskipun *train loss* akhir sedikit lebih tinggi, model dengan DNM mencapai konvergensi lebih cepat dengan akurasi pelatihan tetap tinggi (99,51%). Hal ini menunjukkan bahwa integrasi DNM mempercepat pelatihan tanpa mengorbankan performa model secara drastis. Selanjutnya, pada Gambar 6 menunjukkan perbandingan konvergensi yang ditinjau dari metrik *validation loss* terhadap jumlah epoch. Model ShuffleNetV2 dengan DNM mengalami penghentian dini dengan nilai *validation loss* akhir sebesar 0.3758, sementara model ShuffleNetV2 standar berhenti dengan *validation loss* yang lebih rendah, yaitu 0.1496. Meskipun nilai *validation loss* akhir model yang menggunakan DNM lebih tinggi, model ini menunjukkan kemampuan untuk mencapai konvergensi yang lebih cepat, yang artinya proses pembelajaran lebih efisien. Selain itu, meskipun terdapat fluktuasi nilai *validation loss* pada kedua model, tren umum menunjukkan bahwa model standar mengalami penurunan loss yang lebih stabil setelah

*epoch* ke-10, sedangkan model DNM mengalami sedikit lonjakan pada *epoch ke-11* sebelum kembali menurun. Kedua model sama-sama menunjukkan pemicu *early stopping*, yang ditandai dengan lingkaran berwarna, mengindikasikan bahwa tidak terjadi peningkatan signifikan pada *validation loss* setelah titik tersebut. Sehingga dalam penelitian ini, kombinasi DNM pada ShuffleNetV2 menunjukkan percepatan konvergensi di awal pelatihan (*early convergence*), namun dengan sedikit penurunan akurasi pada data validasi (*generalization gap*) dengan ditandai terdapat lonjakan *validasi loss* di akhir *epoch* optimal. *Trade-off* ini wajar terjadi karena pendekatan dendritik cenderung mempercepat pembelajaran fitur lokal [10].



Gambar 5. Perbandingan Konvergensi di *Training Loss*



Gambar 6. Perbandingan Konvergensi di *Validation Loss*

Untuk informasi lengkap hasil dari grafik diatas disajikan pada Tabel 5 yang meliputi nilai akhir *training loss* (Train Loss), *training accuracy* (Train Acc), *validation loss* (Val Loss), dan *validation accuracy* (Val Acc), serta total epoch digunakan tiap model.

TABEL 5  
KOMPARASI EVALUASI PELATIHAN

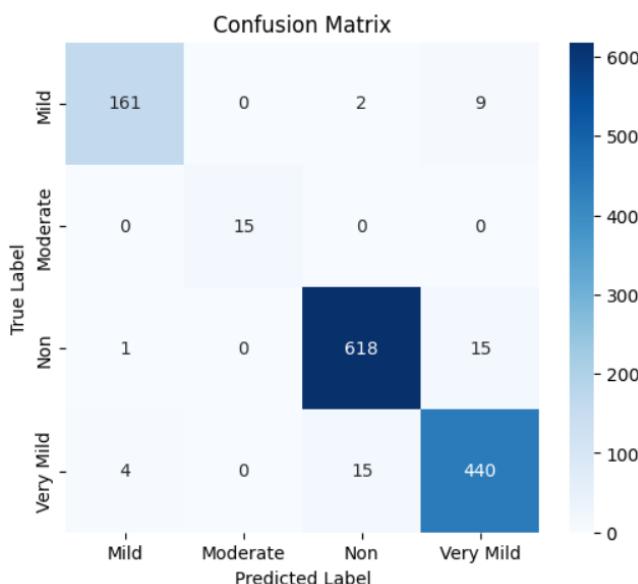
Model	Train Acc(%)	Val Acc(%)	Train Loss	Val Loss	Epoch
ShuffleNetV2	<b>99,98</b>	<b>96,00</b>	<b>0.002</b>	<b>0.15</b>	22
dendritic ShuffleNetV2	99,51	87,60	0.016	0.38	<b>15</b>

Berdasarkan hasil di atas, ShuffleNetV2 dengan integrasi DNM (*dendritic* ShuffleNetV2) menunjukkan potensi sebagai alternatif yang efisien untuk melakukan klasifikasi AD. Selain ditunjukkan melalui grafik konvergensi yang menggambarkan kemampuan model dalam mencapai pelatihan optimal dalam waktu yang lebih singkat, evaluasi lanjutan juga dilakukan melalui proses uji. Hasil dari pengujian tersebut divisualisasikan dalam bentuk *confusion matrix*, yang memberikan gambaran lebih rinci terkait performa model dalam mengklasifikasikan masing-masing kelas secara akurat.

Berdasarkan *confusion matrix* pada Gambar 7 yang menunjukkan hasil pengujian model ShuffleNetV2 yang telah diintegrasikan dengan DNM, model menunjukkan performa klasifikasi yang cukup baik. Performa terbaik ditunjukkan pada kelas *Non Demented (Non)* dan *Very Mild Demented (Very Mild)*. Sebanyak 618 sampel *Non Demented* berhasil diklasifikasikan dengan benar, meskipun terdapat 15 sampel yang keliru diklasifikasikan sebagai *Very Mild*. Untuk kelas *Very Mild*, model berhasil mengklasifikasikan 440 sampel dengan tepat, sementara 15 sampel diklasifikasikan sebagai *Non Demented* dan 4 sampel sebagai *Mild Demented (Mild)*.

Pada kelas *Mild*, model klasifikasi 161 sampel dengan benar, namun terjadi kesalahan pada 11 sampel yang tersebar ke kelas *Non Demented* dan *Very Mild*. Adapun pada kelas *Moderate Demented (Moderate)*, model menunjukkan akurasi sempurna dengan seluruh 15 sampel berhasil diklasifikasikan dengan benar tanpa kesalahan.

Secara keseluruhan, *confusion matrix* tersebut menggambarkan bahwa model memiliki kemampuan generalisasi yang baik, khususnya dalam membedakan antara individu non-demensia dan mereka yang berada pada tahap awal demensia. Meskipun begitu, masih terdapat tantangan dalam membedakan sampel dengan gejala ringan (*Mild*) dan sangat ringan (*Very Mild*).



Gambar 7. *Confusion Matrix* Dendritic ShuffleNetV2

Melalui hasil *confusion matrix*, diperoleh nilai *precision*, *recall*, dan *F1-Score* dari model untuk setiap kelas yang ada. Nilai evaluasi tersebut dapat dilihat pada Tabel 6.

TABEL 6  
HASIL EVALUASI TIAP KELAS KLASIFIKASI

Class	Pre	Rec	F1
Mild	0.9699	0.9360	0.9527
Moderate	1.0000	1.0000	1.0000
Non	0.9732	0.9748	0.9740
Very Mild	0.9483	0.9586	0.9534

#### IV. KESIMPULAN

Berdasarkan hasil percobaan dan analisa yang telah dilakukan, integrasi DNM ke dalam arsitektur ShuffleNetV2 mampu meningkatkan metrik klasifikasi pada dataset Alzheimer MRI dengan jumlah dendrit (M) tertentu. Dibandingkan ShuffleNetV2 biasa, *Dendritic* ShuffleNetV2 menunjukkan kenaikan rata-rata akurasi sebanyak ~0,79%, *precision*, *recall*, dan *F1-score* pada weighted average masing-masing sebesar ~0,8%. Hal ini menegaskan potensi DNM untuk menyerap informasi non-linier yang tidak ditangkap oleh neuron standar. Penambahan lapisan DNM hanya menambah parameter sebesar ~1,6% ( $2,48\text{ M} \rightarrow 2,52\text{ M}$ ), sementara FLOPs tetap sama ( $0,31\text{ G}$ ), sehingga model masih layak dijalankan pada perangkat edge dengan keterbatasan memori dan komputasi. Selain itu, model DNM mencapai konvergensi lebih cepat (15 epoch vs. 22 epoch), menandakan efisiensi dalam proses pelatihan. Analisis *confusion matrix* menunjukkan bahwa kelas *Mild Demented* dan *Very Mild Demented* masih saling tumpang tindih. Meskipun memiliki performa baik, akurasi per kelas untuk membedakan gejala ringan memerlukan peningkatan. Meskipun model DNM memiliki nilai *validation loss* akhir yang sedikit lebih tinggi, stabilitas pelatihan dan kemampuan generalisasinya cukup baik, terutama dalam membedakan kelas mayoritas (*Non Demented*) dan tahap awal demensia. *Early stopping* di masing-masing model juga menunjukkan tidak ada tanda signifikan *overfitting* berlebih.

#### DAFTAR PUSTAKA

- [1] A. G. M. Sianturi, "Stadium, Diagnosis, dan Tata laksana Penyakit Alzheimer," *Majalah Kesehatan Indonesia*, vol. 2, no. 2, pp. 39–44, Oct. 2021, doi: 10.47679/makein.202132.
- [2] Y. Zhao *et al.*, "Application of Deep Learning for Prediction of Alzheimer's Disease in PET/MR Imaging," *Bioengineering*, vol. 10, no. 10, pp. 1–17, Oct. 2023, doi: 10.3390/bioengineering10101120.
- [3] J. S. Purba, "Inflamasi dalam Patologi Penyakit Alzheimer," *MEDICINUS*, vol. 33, no. 3, pp. 65–71, Dec. 2020, doi: 10.56951/MEDICINUS.V33I3.70.
- [4] K. S. Frederiksen *et al.*, "Focusing on earlier diagnosis of Alzheimer's disease," *Future Neurol*, vol. 19, no. 1, 2024, doi: 10.2217/fnl-2023-0024.
- [5] Y. S. Austin *et al.*, "Klasifikasi Penyakit Alzheimer Dari Scan MRI Otak Menggunakan Convnext," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 6, pp. 1223–1232, Dec. 2024, doi: 10.25126/jtiik.2024118117.

- [6] V. P. Nithya, N. Mohanasundaram, and R. Santhosh, "An Early Detection and Classification of Alzheimer's Disease Framework Based on ResNet-50," *Current Medical Imaging Formerly Current Medical Imaging Reviews*, vol. 20, Sep. 2023, doi: 10.2174/1573405620666230825113344.
- [7] Q. Zhang, Y. Long, H. Cai, and Y.-W. Chen, "Lightweight Neural Network for Alzheimer's Disease Classification Using Multi-slice sMRI," *Magn Reson Imaging*, vol. 107, pp. 164–170, Apr. 2024, doi: 10.1016/J.MRI.2023.12.010.
- [8] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11218 LNCS, pp. 122–138, Jul. 2018, doi: 10.48550/arXiv.1807.11164.
- [9] X. Wen, M. Zhou, A. Albeshri, L. Huang, X. Luo, and D. Ning, "Improving Classification Performance in Dendritic Neuron Models through Practical Initialization Strategies," *Sensors*, vol. 24, no. 6, Mar. 2024, doi: 10.3390/s24061729.
- [10] S. Chavlis and P. Poirazi, "Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning," *Nature Communications* 2025 16:1, vol. 16, no. 1, pp. 1–17, Apr. 2024, <https://doi.org/10.1038/s41467-025-56297-9>.
- [11] Q. DU, Z. LIU, Y. SONG, N. WANG, Z. JU, and S. GAO, "A Lightweight Dendritic ShuffleNet for Medical Image Classification," *IEICE Trans Inf Syst*, p. 2024EDP7059, 2025, doi: 10.1587/TRANSINF.2024EDP7059.
- [12] S. Chavlis and P. Poirazi, "Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning," *Nat Commun*, vol. 16, pp. 1–17, Jan. 2025, doi: 10.1038/s41467-025-56297-9.
- [13] I. S. Jones and K. P. Kording, "Can Single Neurons Solve MNIST? The Computational Power of Biological Dendritic Trees," Sep. 2020, [Online]. Available: <http://arxiv.org/abs/2009.01269>
- [14] H. Yu, J. Shi, J. Qian, S. Wang, and S. Li, "Single Dendritic Neural Classification with an Effective Spherical Search-Based Whale learning algorithm," *Mathematical Biosciences and Engineering*, vol. 20, no. 4, pp. 7594–7632, 2023, doi: 10.3934/mbe.2023328.
- [15] N. Wang, Z. Yuan, Y. Gao, R.-L. Wang, and S. Gao, "Dendritic Aggregated Residual Deep Learning for Meningioma MRI Diagnosis," *IEICE Trans Inf Syst*, p. 2024EDL8049, 2025, doi: 10.1587/TRANSINF.2024EDL8049.
- [16] Y. Ding, J. Yu, C. Gu, S. Gao, and C. Zhang, "A Multi-In and Multi-Out Dendritic Neuron Model and its Optimization," *Knowl Based Syst*, vol. 286, Sep. 2023, doi: 10.1016/j.knosys.2024.111442.
- [17] X. Qian, C. Tang, Y. Todo, Q. Lin, and J. Ji, "Evolutionary Dendritic Neural Model for Classification Problems," *Complexity*, vol. 2020, 2020, doi: 10.1155/2020/6296209.