

# **MONOGRAF: KOMPARASI MODEL TRANSFER LEARNING ALGORITMA CNN PADA PENYAKIT ALZHEIMER**

PURWONO, S.KOM., M.KOM.  
ALFIAN MA ARIF S.T., M.ENG.  
DR. ISWANTO, S.T., M.ENG., IPM  
KHOIRUN NISA, S.KOM., M.KOM.



**Penerbit UHB Press**

# **Monograf: Komparasi Model Transfer Learning Algoritma CNN pada Penyakit Alzheimer**

Purwono, S.Kom., M.Kom.  
Alfian Ma Arif S.T., M.Eng.  
Dr. Iswanto, S.T., M.Eng., IPM.  
Khoirun Nisa, S.Kom., M.Kom.



**Penerbit UHB Press**

# **Monograf: Komparasi Model Transfer Learning Algoritma CNN pada Penyakit Alzheimer**

Oleh:

Purwono, S.Kom., M.Kom.  
Alfian Ma Arif S.T., M.Eng.  
Dr. Iswanto, S.T., M.Eng., IPM  
Khoirun Nisa, S.Kom., M.Kom.

Hak Cipta © 2022 pada penulis,

Editor: Dewira Barlian Geasil

Hak Cipta dilindungi oleh undang-undang. Dilarang memperbanyak atau memindahkan Sebagian atau keseluruhan isi buku ini dalam bentuk apapun, secara elektronik maupun mekanis, termasuk mefotokopi, merekam, atau dengan Teknik perekaman lainnya, tanpa izin tertulis dari penerbit.

Diterbitkan oleh **Penerbit UHB Press**

Jl. Raden Patah No.100, Ledug, Kembaran, Banyumas,  
Jawa Tengah, Telp. (0281) 6843493, Fax. (0281) 6843494,  
Purwokerto 53182

**Purwono, S.Kom., M.Kom,  
Alfian Ma Arif S.T., M.Eng.,  
Dr. Iswanto, S.T., M.Eng., IPM  
Khoirun Nisa, S.Kom., M.Kom**

-edisi Pertama – Purwokerto: UHB Press, 2022

xvi + - 77 hlm, 1 Jil: 23 cm

**ISBN: 978-623-88102-3-9**



## Sinopsis

Penyakit Alzheimer dapat diartikan sebagai kondisi neurodegeneratif progresif yang diikuti dengan kondisi penurunan kejiwaan, kognitif dan struktural yang menyumbang 60% - 80% pada semua kasus demensia. Penyakit ini dapat didiagnosis dengan menggunakan pemeriksaan pencitraan, penilaian klinik dan tes neuropsikologis.

Salah satu perkembangan teknologi yang dapat digunakan untuk mendiagnosa penyakit alzheimer dengan pencitraan ialah *deep learning*. Buku ini membahas bagaimana cara untuk menggunakan algoritma CNN dalam melakukan klasifikasi pada penyakit alzheimer dengan berfokus pada pemilihan model transfer learning yang digunakan.

Buku ini fokus pada pembangunan model CNN yang meliputi pemrosesan dataset, pembagian data latih dan uji, pembuatan model transfer learning CNN seperti InceptionV3, Inception ResNet, VGG16, ResNet50 dan Exception hingga proses untuk mengevaluasi model yang sudah dibuat.

# KATA PENGANTAR

*Assalamualaikum warahmatullahi wabarakatuh*

Puji dan syukur Penulis panjatkan kehadiran Allah SWT, atas berkah, rahmat dan hidayah-Nya sehingga penulisan buku monograf dengan judul Komparasi Model Transfer Learning Algoritma CNN pada Penyakit Alzheimer dapat diselesaikan. Monograf ini dapat selesai tidak lepas karena dukungan banyak pihak seperti keluarga, teman-teman dosen informatika. Untuk itu Penulis mengucapkan terimakasih yang sedalam-dalamnya kepada pihak-pihak yang telah membantu. Buku ini fokus pada pembangunan model CNN yang meliputi pemrosesan *dataset*, pembagian data latih dan uji, pembuatan model *transfer learning* CNN seperti InceptionV3, Inception ResNet, VGG16, ResNet50 dan Exception hingga proses untuk mengevaluasi model yang sudah dibuat. Penulis berharap buku monograf ini dapat memberikan manfaat kepada mahasiswa informatika ataupun sejawat dimanapun berada. Akhir kata, Penulis menyadari buku monograf ini masih jauh dari kata sempurna, sehingga saran dan kritik yang bersifat membangun sangat diharapkan oleh Penulis untuk perbaikan penulisan monograf ini.

*Wassalamualaikum warahmatullahi wabarakatuh.*

Purwokerto, Desember 2022

Penulis

# DAFTAR ISI

Halaman Sampul .....	ii
Sinopsis .....	iv
KATA PENGANTAR.....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR.....	vii
DAFTAR TABEL .....	viii
BAB I Pendahuluan.....	1
Penyakit Alzheimer .....	1
<i>Artificial Intelligence</i> .....	2
<i>Deep Learning</i> .....	3
<i>Deep Learning</i> di Bidang Kesehatan.....	4
BAB II Deep Learning pada Prediksi Penyakit Alzheimer... 7	7
Penelitian <i>Deep Learning</i> .....	7
Penggunaan Algoritma CNN .....	9
BAB III Convolutional Neural Network .....	11
Layer CNN.....	11
Loss Function.....	18
Transfer Learning.....	20
BAB IV Pembuatan Model Klasifikasi CNN Alzheimer.... 24	24
Deskripsi Dataset .....	25
Pemrosesan Data .....	26
Evaluation Metriks .....	27
BAB V Hasil Model Klasifikasi CNN .....	30
Model Pembelajaran.....	30
Model Pengujian .....	47
PENUTUP .....	51
DAFTAR PUSTAKA .....	52

# DAFTAR GAMBAR

Gambar 1 Metodologi Penelitian.....	24
Gambar 2 Dataset alzheimer .....	25
Gambar 3 Arsitektur CNN .....	11
Gambar 4 Piksel Sebuah Gambar.....	11
Gambar 5 Convolutional Layer .....	13
Gambar 6 Pooling Layer .....	14
Gambar 7 Fully Connected Layer .....	15
Gambar 8 Tren metrik selama pelatihan.....	46
Gambar 9 Hasil Komparasi Transfer Learning.....	47
Gambar 10 Hasil Akurasi Pengujian Model.....	49

## DAFTAR TABEL

Table 1 Confusion Matrix .....	27
Table 2 Hasil Pembelajaran Model .....	45
Table 3 Hasil Evaluasi Model .....	48
Table 4 Hasil Model Pelatihan .....	49



# BAB I

## Pendahuluan

### Penyakit Alzheimer

Penyakit Alzheimer (AD) dapat diartikan sebagai kondisi neurodegeneratif progresif yang diikuti dengan kondisi penurunan kejiwaan, kognitif dan struktural yang menyumbang 60% - 80% pada semua kasus demensia (Mirabnahrizam et al. 2022). Ciri lain dari alzheimer ialah penurunan progresif dalam ingatan, kemampuan visuo spasial yang dapat mengganggu fungsi sosial (Ota et al. 2022). Penyakit ini memiliki ciri patologis klasik berupa plak amiloid dan kekusutan *neurofibrillary* (Katsumata et al. 2022).

Populasi yang menua dikaitkan sebagai salah satu penyebab terjadinya peningkatan prevalensi penyakit Alzheimer dan demensia di seluruh dunia (de Oliveira Silva et al. 2019). Hingga tahun 2040, jumlah orang yang diprediksi terkena penyakit demensia sebanyak 81,1 juta jiwa (de Oliveira Silva et al. 2019). Penyakit ini berpotensi terus meningkat dari 10 juta per tahun hingga 13 juta per tahun pada tahun 2024 jika tidak adanya tindakan pengendalian (Tao et al. 2020).

*Magnetic resonance imaging* (MRI) telah mengungkapkan perubahan struktural dan fungsional otak dalam perkembangan penyakit Alzheimer. Secara khusus, penipisan kortikal regional dan kehilangan volume diperkirakan akibat dari proses neurodegeneratif dan biasanya terjadi setelah deposisi amiloid dan agregasi tau *hiperfosforilasi*. Selain itu, perbedaan dalam konektivitas fungsional keadaan

istirahat (*resting-state*) dianggap mencerminkan gangguan jaringan otak, terutama di daerah dengan deposisi amiloid (Millar et al. 2022).

Sekitar 80% risiko penyakit Alzheimer diwariskan, namun mekanisme biologis yang mendasari heritabilitas masih kurang dipahami (Katsumata et al. 2022). Pada penyakit Alzheimer (AD), disregulasi serebrovaskular adalah faktor penyerta utama dan merupakan salah satu penanda awal patofisiologi AD. Penurunan fungsi serebrovaskular diamati baik pada pasien dengan AD maupun individu muda dengan peningkatan risiko demensia. Ini secara luas menunjukkan bahwa perubahan fungsi serebrovaskular merupakan faktor risiko untuk AD, bukan konsekuensi dari penyakit, yang mungkin ada sepanjang umur seseorang (Chandler et al. 2022).

AD dapat didiagnosis dengan menggunakan pemeriksaan pencitraan, penilaian klinik dan tes neuropsikologis (Lin et al. 2022). Ternyata pemeriksaan pencitraan dengan model *deep learning* untuk mendiagnosis penyakit ini telah menarik banyak perhatian penelitian (Nguyen et al. 2022). Terdapat juga upaya penelitian dalam mengeksplorasi teknik pencitraan otak fungsional untuk diagnosis dini dan evaluasi kemanjuran pengobatan (Khan and Zubair 2022). Metode *deep learning* umum diterapkan untuk mendeteksi penyakit AD dari data *neuroimaging* (Ebrahimighahnavieh, Luo, and Chiong 2020).

### ***Artificial Intelligence***

Kecerdasan buatan (AI) adalah disiplin ilmu yang berusaha untuk mereproduksi cara-cara yang

mirip manusia dalam memahami, bernalar, belajar, dan memecahkan masalah dan merupakan bidang yang diminati dalam aplikasi klinis. AI memperluas teknik statistik tradisional, memungkinkan penggalan informasi untuk mendukung pengambilan keputusan dan penelitian (Martinez-Millana et al. 2022).

AI didasarkan pada algoritme matematika, dan akurasi prediksi bergantung pada algoritme dan kualitas data. Belakangan ini, penggunaan AI juga telah menyebar ke dunia kedokteran, di mana AI telah diterapkan dalam skrining, perhitungan risiko penyakit, triase, pengembangan obat, pengobatan, pemantauan, dan interpretasi gambar (Sarno et al. 2023).

Pembelajaran mesin (*Machine Learning*) dan pembelajaran mendalam (*Deep Learning*) keduanya adalah jenis AI. Singkatnya, pembelajaran mesin adalah AI yang dapat beradaptasi secara otomatis dengan sedikit campur tangan manusia. Pembelajaran mendalam adalah bagian dari pembelajaran mesin yang menggunakan jaringan saraf tiruan untuk meniru proses pembelajaran otak manusia (Coursera 2022).

### ***Deep Learning***

Istilah *Artificial Intelligence* (AI) mengacu pada seluruh jagad teknologi komputasi yang menunjukkan sesuatu yang menyerupai kecerdasan manusia. Dengan kata lain, AI adalah usaha membangun artefak cerdas. Definisi AI yang paling diterima adalah yang dikemukakan oleh Alan Turing, sebuah komputer dapat dikatakan memiliki AI jika

dapat meniru respon manusia dalam kondisi tertentu (Oliveira and Bollen 2023).

Kemajuan terbaru dalam aplikasi berbasis kecerdasan buatan memiliki dampak yang kuat di bidang ini. Mereka telah berkontribusi pada kemajuan signifikan visi komputer (*computer vision*), pembelajaran mesin (*machine learning*), dan solusi pembelajaran mendalam (*deep learning*) dalam pengembangan sistem otomatis dan kuat (Coulibaly et al. 2022). Pembelajaran mendalam telah membuat banyak kemajuan teknis saat ini. Namun demikian, interpretasi hasil model pelatihan pada data itelah menimbulkan perdebatan panjang di komunitas ilmiah dan pengguna akhir.

### ***Deep Learning* di Bidang Kesehatan**

Teknik dan algoritme *deep learning* muncul sebagai teknologi yang mengganggu dengan potensi untuk mengubah ekonomi, lingkungan, dan masyarakat global (Fu et al. 2022). Penerapan pendekatan *deep learning* pada bidang kesehatan menjadi lebih menarik dikarenakan adanya peningkatan kompleksitas data layanan kesehatan (Shamshirband et al. 2021). Data kesehatan juga terus bertambah seiring adanya ledakan data pasien dari sumber data yang beragam dalam berbagai format sehingga dapat diolah dengan model, algoritme, alat, dan teknik kecerdasan komputasi untuk menuai wawasan dan pola layanan kesehatan yang berharga (Thirunavukarasu et al. 2022). *Big data* telah menjadi teknik yang akrab digunakan dalam bidang kesehatan karena memiliki kumpulan data berguna yang sangat banyak, cepat dan sangat sulit diolah oleh tenaga

medis dengan bantuan alat yang ada (Pustokhin et al. 2021).

*Deep Learning* yang telah muncul sebagai alat yang efektif untuk menganalisis big data menggunakan algoritme kompleks dan jaringan syaraf tiruan untuk melatih mesin/komputer sehingga mereka dapat belajar dari pengalaman, mengklasifikasikan, dan mengenali data/gambar seperti halnya otak manusia. Dalam *Deep Learning*, *Convolutional Neural Network* atau CNN adalah jenis jaringan saraf tiruan, yang banyak digunakan untuk pengenalan dan klasifikasi gambar/objek. *Deep Learning* dengan demikian mengenali objek dalam gambar dengan menggunakan CNN. CNN memainkan peran utama dalam beragam tugas/fungsi seperti masalah pemrosesan gambar, tugas visi komputer seperti lokalisasi dan segmentasi, analisis video, untuk mengenali hambatan dalam mobil yang dapat mengemudi sendiri, serta pengenalan ucapan dalam pemrosesan bahasa alami. Karena CNN memainkan peran penting dalam area yang berkembang pesat dan berkembang pesat ini, CNN sangat populer di *Deep Learning*.

Popularitas CNN juga telah merambah pada implementasi pada bidang kesehatan. Sebagai contoh CNN pernah digunakan untuk mendeteksi *pneumonia* pada anak dan telah terbukti mampu mencapai kinerja yang luar biasa dalam tugas segmentasi, analisis, dan klasifikasi citra medis (Vrbančić and Podgorelec 2022). Contoh lain ialah penelitian tentang deteksi parasit malaria menggunakan teknik algoritma *deep learning* berbasis CNN dengan hasil kinerja yang maksimal (Alnussairi and Ibrahim 2022). CNN telah digunakan secara maksimal pada bidang perawatan

kesehatan (Ein Shoka et al. 2022). Model CNN mampu diterapkan sebagai alat diagnostik otomatis untuk mendukung para ahli dalam mendeteksi beberapa penyakit.

## BAB II

# Deep Learning pada Prediksi Penyakit Alzheimer

### Penelitian *Deep Learning*

Prediksi AD menggunakan CNN merupakan pekerjaan yang cukup menantang. Beberapa penelitian terkait penggunaan CNN dalam klasifikasi AD dirangkum pada bagian ini. Kami menemukan berbagai jenis penelitian terkait di antaranya telah dilakukan oleh Kruthika (Kale 2019) yang mengusulkan model 3D *Capsule Network*, 3D-CNN dan teknologi 3D-*autoencoder* pra-pelatihan untuk deteksi dini AD. Model ini berhasil menghasilkan nilai akurasi hingga 98,42% dalam klasifikasi penyakit AD.

Penelitian yang telah dilakukan oleh Geenedy (EL-Geenedy et al. 2023) mengembangkan alur berbasis pembelajaran yang mendalam dengan algoritma CNN untuk diagnosis dan stratifikasi tahapan AD yang akurat. Model ini menghasilkan akurasi pengujian keseluruhan hingga 99,68%.

Penelitian yang telah dilakukan Liu (Liu et al. 2020) mengusulkan kerangka pembelajaran mendalam multi-model menggunakan algoritma CNN untuk segmentasi *hippocampal* otomatis bersama dan klasifikasi AD menggunakan data MRI struktural. Model ini mampu mencapai nilai akurasi hingga 88,9% dan AUC sebesar 92,5%.

Penelitian yang telah dilakukan oleh Bringas (Bringas et al. 2020) mengidentifikasi stadium pasien AD melalui penggunaan data mobilitas dan model

deep learning CNN. Metode berbasis CNN mencapai akurasi 90,91% dan skor F1 0,897, dianggap sangat meningkatkan hasil yang diperoleh oleh pengklasifikasi berbasis fitur tradisional.

Penelitian yang telah dilakukan oleh Poloni (Poloni and Ferrari 2022) membuat model ansambel hippocampal CNN yang dalam untuk estimasi usia otak yang diterapkan pada diagnosis Alzheimer. Hasil dari penelitian ini ialah dibentuknya kerangka estimasi usia yang efisien hanya dengan menggunakan daerah *hippocampal* yang mengeksplorasi hubungan kesalahan prediksi usia otak dari subjek kognitif normal yang cocok dengan usia dengan subjek AD dan MCI. Penelitian ini telah mengembangkan dua jaringan saraf konvolusional. Yang pertama mencapai metrik canggih yang sangat kompetitif, yaitu *mean absolute error* (MAE) sebesar 3,31 dan *root mean square error* (RMSE) sebesar 4,65.

Penelitian yang telah dilakukan oleh Abbas (Qasim Abbas, Chi, and Chen 2023) membuat Jaringan saraf *convolutional* domain yang diubah untuk diagnosis penyakit Alzheimer menggunakan MRI struktural. Penelitian ini mengusulkan jaringan saraf *convolutional Jacobian domain* tiga dimensi baru (JD-CNN) untuk mendiagnosis subjek AD dan mencapai kinerja klasifikasi yang sangat baik tanpa keterlibatan kerangka kerja deteksi LM. JD-CNN yang diusulkan menunjukkan kinerja klasifikasi yang unggul untuk teknik canggih yang dilaporkan sebelumnya.

Penelitian yang dilakukan oleh Alorf (Alorf and Khan 2022) menganalisis dan menyajikan metode klasifikasi multi-label enam stadium



Alzheimer menggunakan rs-fMRI dan *deep learning*. Model yang diusulkan memecahkan masalah klasifikasi multi-kelas dengan mengekstraksi jaringan konektivitas fungsional otak dari data rs-fMRI dan menggunakan dua pendekatan pembelajaran mendalam, *Stacked Sparse Autoencoder* dan *Brain Connectivity Graph Convolutional Network*. Hasil model yang disarankan dinilai menggunakan pendekatan k-fold cross-validation, dan akurasi rata-rata 77,13% dan 84,03% dicapai untuk klasifikasi multi-label masing-masing menggunakan *Stacked Sparse Autoencoder* dan *Brain Connectivity Based Convolutional Network*.

### **Penggunaan Algoritma CNN**

*Deep Neural Networks* (DNNs) telah menciptakan perspektif baru untuk *Computer Vision* dan baru-baru ini diterapkan untuk *Content-Based Image Retrieval* (CBIR). Perubahan struktural praklinis yang dapat diamati memberikan peluang untuk deteksi dini AD menggunakan alat klasifikasi gambar, seperti jaringan saraf convolutional (CNN) dan Jaringan Kapsul (Kruthika, Rajeswari, and Maheshappa 2019).

Selama dekade terakhir, *deep convolutional neural network* (CNNs) telah menjadi bahan pokok untuk klasifikasi AD dari gambar otak yang diperoleh dengan *magnetic resonance imaging* (MRI) (Wolf, Pölsterl, and Wachinger 2022). MRI digunakan sebagai media diagnosis awal penyakit ini. Beberapa contoh tindakan non-invasif, seperti pencitraan resonansi magnetik struktural (sMRI), pencitraan resonansi magnetik fungsional (fMRI), dan tomografi

emisi positron (PET), dll., tersedia untuk diagnosis AD (Qasim Abbas et al. 2023).

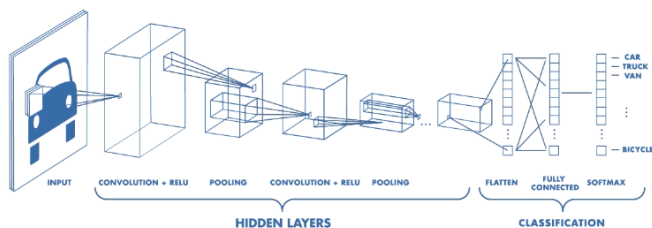
CNN memiliki berbagai jenis model *transfer learning* yang sering digunakan oleh para peneliti pemrosesan *image* seperti VGG-16, ResNet-50, dan DenseNet-201 (Gonçalves, Souza, and Fernandes 2022). Selain itu juga terdapat model *transfer learning* CNN lainnya seperti VGG19, InceptionV3, DenseNet121, MobileNet, dan Xception (Thepade et al. 2022). Merancang CNN adalah tugas yang kompleks dan membutuhkan pengetahuan ahli untuk mengoptimalkan kinerja dan arsitektur jaringan (Aradhya et al. 2022).

# BAB III

## Convolutional Neural Network

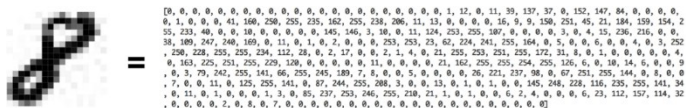
### Layer CNN

CNN memiliki empat jenis layer antara lain *convolution layer*, *pooling layer*, *fully connected layer* dan *non linearity layer* (Bezdan and Baćanin Džakula 2019). Ilustrasi dari keempat layer tersebut dapat dilihat pada Gambar 3 (The Mathworks 2018). Keempat layer tersebut akan diperjelas pada sub bagian di bawah ini.



Gambar 1 Arsitektur CNN

Sebuah gambar akan dilihat oleh sebuah mesin dalam bentuk kumpulan angka-angka atau bisa kita sebut dengan matriks. Setiap angka tersebut menandakan sebuah intensitas cahaya pada sebuah titik yang disebut dengan pixel. Ilustrasi *pixel* dari sebuah gambar milik Adam geiytgey dalam website medium yang dapat dilihat pada Gambar 4.



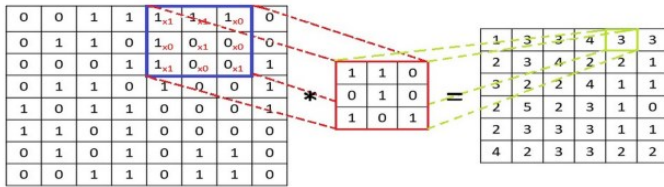
Gambar 2 Pikel Sebuah Gambar

## Convolutional Layer

*Convolutional layer* menggunakan filter kernel untuk menghitung operasi konvolusi dari gambar input untuk mengekstrak fitur fundamental. Konvolusi adalah operasi matematika yang memungkinkan penggabungan dua set informasi. Dalam kasus CNN, konvolusi diterapkan pada input data untuk memfilter informasi dan menghasilkan peta fitur (Ghosh et al. 2019). *Filter kernel* memiliki dimensi yang sama namun untuk parameter konstan bernilai lebih kecil dibandingkan dengan gambar *input* (Singh, Meitei, and Majumder 2020). Kernel hanyalah filter yang digunakan untuk mengekstraksi fitur dari gambar. Kernel adalah matriks yang bergerak di atas data input, melakukan produk titik dengan sub-region data input, dan mendapatkan output sebagai matriks produk titik. Kernel bergerak pada input data dengan nilai langkahnya. Jika nilai langkahnya adalah 2, maka kernel bergerak sebanyak 2 kolom piksel dalam matriks masukan. Singkatnya, kernel digunakan untuk mengekstraksi fitur tingkat tinggi seperti tepian dari gambar (Ghosh et al. 2019). Sebagai contoh, perhitungan dari gambar *scalogram* dengan ukuran  $35 \times 35 \times 35$  2D, ukuran filter yang dapat diterima adalah  $f \times f \times 2$ , dimana  $f = 3, 5, 7$  hingga selanjutnya. Namun perlu diketahui bahwa ukuran filter harus lebih kecil daripada gambar input. Filter mask meluncur keseluruhan gambar input step by step dan memperkirakan titik produk antara bobot filter kernel dengan nilai gambar yang diinputkan. Hal ini akan menghasilkan nilai peta aktivasi 2D. CNN akan mempelajari fitur visual yang dimiliki oleh sebuah gambar. Persamaan *convolutional layer* umum dapat ditampilkan sebagai berikut. Gambar 5

merupakan ilustrasi sederhana bagaimana proses dari *activation map* komputasi dari CNN.

$$\begin{aligned}
 \text{Activation map} &= \text{Input} * \text{Filter} \\
 &= \sum_{y=0}^{\text{columns}} \left( \sum_{x=0}^{\text{rows}} \right) \text{Input}(x - p, y \\
 &\quad - q) \text{Filter}(x, y) \quad (1)
 \end{aligned}$$



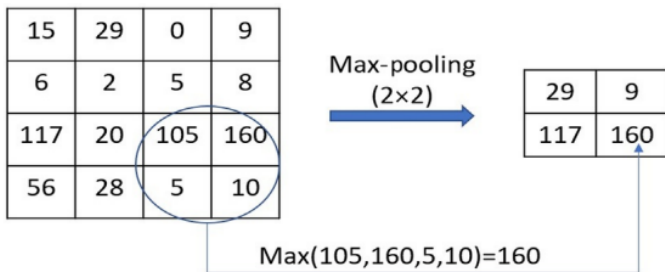
Gambar 3 Convolutional Layer

*Convolutional* memiliki tiga parameter yang digunakan untuk mendefinisikan layer convolusi. Ketiga layer tersebut adalah *Kernel Size*, *Stride Length* dan *Padding* (Suha and Sanam 2022). Kernel size merupakan ukuran *kernel* atau bisa disebut dengan *kernel* geser (Ding et al. 2018). Stride length ialah berapa banyak kernel yang meluncur sebelum membuat titik produk untuk menghasilkan output pixel (Riad et al. 2022). Padding ialah ukuran *frame* nol yang disiapkan di sekitar peta fitur input (Nguyen et al. 2019).

### Pooling Layer

*Pooling layer* akan menggabungkan antara dua layer convolusi yang berurutan. *Pooling layer* mengurangi jumlah parameter dan komputasi dengan representasi *down-sampling*. Fungsi *pooling* ini bisa

mendapatkan hasil yang maksimal atau rata-rata. Penggabungan secara maksimum sering kali digunakan karena dapat berfungsi lebih baik (Ke et al. 2018). *Pooling layer* dapat berguna juga untuk mengurangi *overfitting* ataupun beban komputasi. Gambar 6 merupakan representasi operasi sederhana untuk mereduksi dimensi *activation map* menggunakan pendekatan *max pooling* (Singh et al. 2020).

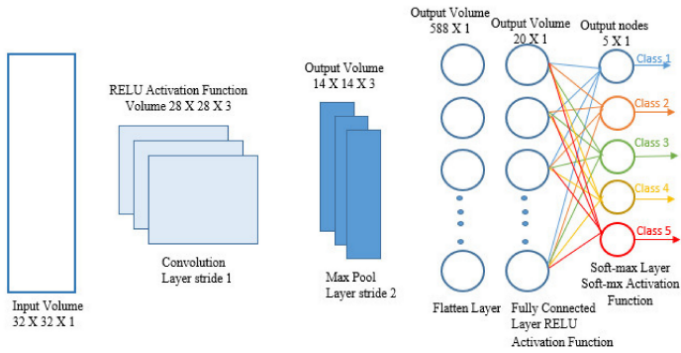


Gambar 4 Pooling Layer

### Fully Connected Layer

Lapisan ketiga merupakan *fully connected* layer atau dapat kita sebut dengan lapisan *output* konvolusi (Bhatt et al. 2021). *Fully connected* tidak lebih dari jaringan saraf umpan-maju seperti yang ditunjukkan pada Gambar 7 *Layer* ini ditemukan di lapisan paling bawah jaringan. *Layer* terhubung untuk menerima input dari penyatuan akhir atau lapisan keluaran lapisan konvolusi, yang diratakan sebelum dikirim sebagai input. Distribusi secara merata dari output berarti membuka gulungan semua nilai dari output yang diperoleh setelah *pooling* atau *convolutional layer* terakhir menjadi vektor (matriks 3D). Cara ini merupakan teknik sederhana untuk mempelajari kombinasi nonlinier tingkat tinggi dari

sebuah fitur yang direpresentasikan oleh *output convolutional layer* (Bhatt et al. 2021).



Gambar 5 Fully Connected Layer

### Non Linearity Layer (Activation Function)

Activation function memainkan peran penting dalam lapisan CNN. Keluaran filternya adalah disediakannya fungsi matematika lain yang disebut dengan fungsi aktivasi (Bhatt et al. 2021). *ReLU*, yang merupakan singkatan dari *Rectified Linear Unit* (Wang et al. 2021), adalah fungsi aktivasi yang paling umum digunakan dalam fitur ekstraksi CNN. Motif utama di balik penggunaan fungsi aktivasi adalah untuk menyimpulkan output jaringan saraf, seperti ya atau tidak. Fungsi aktivasi memetakan nilai output antara 1 hingga 1 atau 0 hingga 1, dan seterusnya. Aktivasi Fungsi dapat dibedakan menjadi dua jenis, yaitu (Bhatt et al. 2021):

1. Fungsi Aktivasi Linier Ini menggunakan fungsi  $F(x) = CY$ . Dibutuhkan masukan dan mengalikannya dengan konstanta  $c$  (bobot setiap neuron), dan menghasilkan sinyal keluaran sebanding dengan masukannya. Fungsi linier bisa lebih baik daripada step function, karena itu hanya memberikan

jawaban ya atau tidak dan bukan jawaban ganda.

2. Fungsi Aktivasi Non-linear Dalam jaringan saraf modern, aktivasi non-linear fungsi digunakan. Mereka memungkinkan model untuk membangun pemetaan yang rumit antara input dan output jaringan, yang sangat penting untuk pembelajaran dan pemodelan kompleks.

Jenis fungsi aktivasi berikut ini paling banyak atau umum digunakan di CNN dan jaringan saraf dalam lainnya antara lain sebagai berikut (Alzubaidi et al. 2021):

1. Sigmoid: Fungsi aktivasi menggunakan bilangan real sebagai masukan dan untuk luaran dibatasi antara 0 dan 1.

$$f(x)_{\text{sigm}} = \frac{1}{1 + e^{-x}} \quad (2)$$

2. Tanh: Sebetulnya tanh mirip dengan sigmoid karena sama sama menggunakan bilangan real sebagai masukan, namun untuk luarannya dibatasi antara -1 hingga 1.

$$f(x)_{\text{tanh}} = \frac{e^x + e^{-x}}{e^x - e^{-x}} \quad (3)$$

3. ReLU: Menjadi fungsi yang paling umum digunakan oleh CNN. Semua masukan diubah menjadi angka positif. ReLU memiliki beban komputasi yang relatif rendah dibandingkan fungsi yang lain.



$$f(x)ReLU = \max(0, x) \quad (4)$$

4. Leaky ReLU: Jika ReLU bertugas menurunkan skala input negatif, *Leaky ReLU* memastikan masukan tidak pernah diabaikan. Fungsi ini digunakan untuk memecahkan masalah *dying* ReLU.

$$f(x)LeaklyReLU = \begin{cases} x, & \text{if } x > 0 \\ mx, & x \leq 0 \end{cases} \quad (5)$$

5. Noisy ReLU: Fungsi ini digunakan untuk melakukan distribusi *gaussian* untuk membuat noisy ReLU.

$$f(x)NoisyReLU = \max(x + Y), \text{ with } Y \sim N(0, \sigma(x)) \quad (6)$$

6. Parametric Linear Units: Fungsi ini sebagian besar mengadopsi konsep *Leaky ReLU*. Perbedaan diantara keduanya sebetulnya terletak pada leak factor yang diperbaharui melalui proses pelatihan model.

$$f(x)ParametricLinear = \begin{cases} x, & \text{if } x > 0 \\ ax, & x \leq 0 \end{cases} \quad (7)$$

CNN memiliki kapasitas yang baik dalam mempelajari representasi gambar (Dambal, Doddanajedevuru, and Gopalakrishna 2022). CNN adalah model yang dapat menghasilkan *feature maps* secara berurutan, memperoleh karakteristik

sederhana dari data seperti simpul dan tepi gambar pada lapisan pertama dan mengelompokannya dalam pola yang lebih kompleks (Bringas et al. 2020). Feature maps ini diperoleh dengan menerapkan operasi konvolusional dengan *kernel* yang dapat dilatih ke input layer (Bringas et al. 2020). Transformasi *non-linier* dan *pooling* merupakan fungsi pelengkap yang membantu jaringan untuk menyatu (Bringas et al. 2020). Sebuah prediksi kemudian dihasilkan berdasarkan featur maps yang diproses (Bringas et al. 2020). Berikut ini beberapa jenis arsitektur CNN yang digunakan dalam penelitian ini.

### **Loss Function**

Kita dapat menghitung kesalahan prediksi yang dihasilkan oleh model CNN pada sampel pelatihan menggunakan beberapa *loss function*. Kesalahan prediksi ini memberi tahu jaringan bagaimana prediksi mereka dari output aktual, dan kemudian kesalahan ini akan dioptimalkan selama proses pembelajaran model CNN. *Loss function* menggunakan dua parameter untuk menghitung kesalahan, parameter pertama adalah perkiraan output model CNN (juga disebut prediksi) dan yang kedua adalah keluaran aktual (juga dikenal sebagai label) (Ghosh et al. 2019). Ada berbagai jenis fungsi kerugian yang digunakan dalam berbagai jenis masalah Arsitektur CNN antara lain sebagai berikut.

## Cross-Entropy

*Cross-entropy* adalah ukuran perbedaan antara dua distribusi probabilitas untuk variabel acak atau rangkaian peristiwa tertentu. Cross-entropy loss, juga disebut fungsi log loss banyak digunakan untuk mengukur kinerja model CNN, yang keluarannya adalah probabilitas  $p \in \{0, 1\}$ . Fungsi ini banyak digunakan sebagai alternatif *loss function* pada kesalahan kuadrat dalam masalah klasifikasi multi-kelas. Fungsi ini menggunakan aktivasi *softmax* di lapisan output untuk menghasilkan output dalam distribusi probabilitas.

## Euclidean Loss Function

*Euclidean Loss Function* juga disebut kesalahan kuadrat rata-rata banyak digunakan dalam masalah regresi. Galat kuadrat rata-rata antara output yang diprediksi  $p \in \mathbb{R}^N$  dan output aktual  $y \in \mathbb{R}^N$  di setiap neuron dari lapisan output CNN didefinisikan sebagai  $H(p, y) = (p - y)^2$ . Jadi, jika ada  $N$  neuron di lapisan keluaran, ia memperkirakan *euclidean loss function* yang didefinisikan dengan rumus berikut.

$$H(p, y) = \frac{1}{2N} \sum_{i=1}^N (p_i - y_i)^2 \quad (8)$$

## Hinge Loss Function

*Hinge loss function* banyak digunakan dalam masalah klasifikasi biner. Fungsi ini digunakan dalam masalah klasifikasi berbasis "margin maksimum", terutama untuk *support vector machines* (SVM). Di

sini, pengoptimal mencoba memaksimalkan margin antara dua kelas target.

## Transfer Learning

### Inception V3

Model *Inception V3* milik CNN ini secara otomatis merubah ukuran gambar menjadi 299 x 299. *Inception V3* mengurangi jumlah jaringan parameter model dan pada kernel konvolusi asimetris digunakan untuk meningkatkan keragaman pada waktu yang sama (Liu et al. 2021). *Inception V3* memiliki *layer batch normalization* (BN) yang dimasukan sebagai pengatur diantaranya pengklasifikasi tambahan dan *fully connected layer*. Dalam model BN ini, metode gradient descent dapat digunakan untuk mempercepat kecepatan pembelajaran dan konvergensi jaringan saraf dalam. Rumus BN dapat dilihat pada rumus berikut (Cao et al. 2021):

$$B = \{X_{1...m}\}, \gamma, \beta \quad (8)$$

$$\{y_i = BN_{\gamma, \beta}(x_i)\} \quad (9)$$

$$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad (10)$$

$$\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (11)$$

$$\hat{X}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (12)$$

$$y_i \leftarrow y\hat{X}_i + \beta = BN_{\gamma,\beta}(x_i) \quad (13)$$

di mana  $x$  adalah nilai aktivasi minimum batch  $B$ ,  $m$  adalah jumlah aktivasi nilai,  $\gamma$  dan  $\beta$  adalah parameter yang dapat dipelajari ( $\gamma$  bertanggung jawab untuk menyesuaikan varian dalam distribusi nilai dan  $\beta$  bertanggung jawab untuk menyesuaikan posisi rata-rata nilai),  $\mu_B$  mewakili nilai rata-rata dalam satu dimensi,  $\sigma_B^2$  adalah standar deviasi di setiap dimensi peta fitur, dan  $\epsilon$  adalah konstanta.

### **Inception ResNetV2**

*Inception ResNet* merupakan kombinasi dari blok *Inception* dan struktur ResNet (Peng et al. 2020). Modul *ResNet* memungkinkan secara primitif melatih jaringan saraf yang lebih dalam dengan cara memperkenalkan residual koneksi. Blok *inception* mampu mendapatkan banyak informasi dari beberapa jenis skala gambar input. *ResNet* dapat memperoleh nilai akurasi dari kedalaman yang jauh meningkat. Kombinasi keduanya dapat meningkatkan kinerja arsitektur CNN (Peng et al. 2020). *Inception resnet* juga dapat memfasilitasi pengoptimalan jaringan dan memecahkan masalah *over-fitting* (Zhang et al. 2021). *Inception resnet* memiliki 164 layer sehingga dapat mengklasifikasikan sekitar 1000 kategori objek.

### **VGG16**

VGG16 merupakan salah satu jenis arsitektur jaringan CNN yang dapat meningkatkan kinerja pembelajaran mendalam menjadi lebih efektif. Modul

ini terdiri dari kernel konvolusi kecil, kernel penyatuan kecil dan *ReLU* (Kong and Cheng 2022). Terdapat 5 lapisan konvolusional, 3 *fully connected* layer dan sebuah *softmax output*. *Max pooling* digunakan untuk memisahkan layer dan fungsi *ReLU* digunakan sebagai unit aktivasi semua hidden layers. Arsitektur ini menyederhanakan struktur jaringan syaraf. Terdapat 7 x 7 x 512 *feature map* yang terhubung sepenuhnya, dan aktivasi softmax dilakukan untuk menghasilkan data pengenalan dari ketiga objek (Kong and Cheng 2022).

### **Resnet50**

Arsitektur ini dibentuk untuk mengatasi kendala yang terjadi dalam pelatihan *deep learning* seperti pemrosesan yang membutuhkan waktu yang cukup lama dan terbatas pada sejumlah lapisan tertentu (Sarwinda et al. 2021). Keunggulan model *ResNets* dibandingkan dengan model arsitektur lainnya adalah performa model ini tidak menurun meskipun arsitekturnya semakin dalam (Sarwinda et al. 2021). Selain itu, perhitungan komputasi dibuat lebih ringan, dan kemampuan untuk melatih jaringan lebih baik. Model *ResNet* diimplementasikan dengan melewati koneksi pada dua hingga tiga lapisan yang berisi *ReLU* dan normalisasi *batch* di antara arsitektur (He et al. 2016).

### **Xception**

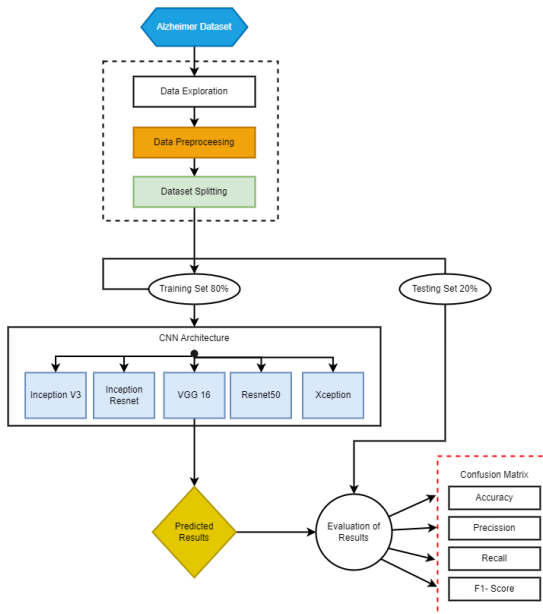
Jaringan *xception* menggantikan konvolusi pada arsitektur *inception v3* dengan *deep separable convolution*, sehingga dapat meningkatkan lebar jaringan dan mengurangi jumlah parameter serta kalkulasi dari model (Chen, Yang, and Zhang 2020).

Pada saat yang sama, mekanise *residual connection* mirip dengan jaringan *ResNet* yang diperkenalkan untuk mempercepat konvergensi jaringan, meningkatkan nilai akurasi klasifikasi hingga meningkatkan kemampuan belajar jaringan ke fitur yang lebih detail (Zhu et al. 2022). Model ini termasuk *input layer*, *convolution layer*, *fully connected layer* dan *output layer*. Dalam model ini, ada lapisan konvolusi yang dapat dipisahkan secara mendalam di lapisan konvolusi, yang terhubung secara linier residu.

# BAB IV

## Pembuatan Model Klasifikasi CNN Alzheimer

Berdasarkan penelitian terkait, kami melihat performa CNN yang sangat bagus dalam penelitian pada AD, sehingga kami mengusulkan penggunaan dari algoritma ini. Secara umum kontribusi kami dalam penelitian ini ialah menggunakan algoritma CNN dalam melakukan klasifikasi pada penyakit alzheimer dengan berfokus pada pemilihan arsitektur yang digunakan. Penelitian ini akan membandingkan berbagai jenis arsitektur populer dari CNN yang memiliki kinerja paling optimal. Alur penelitian ditunjukkan pada Gambar 1.

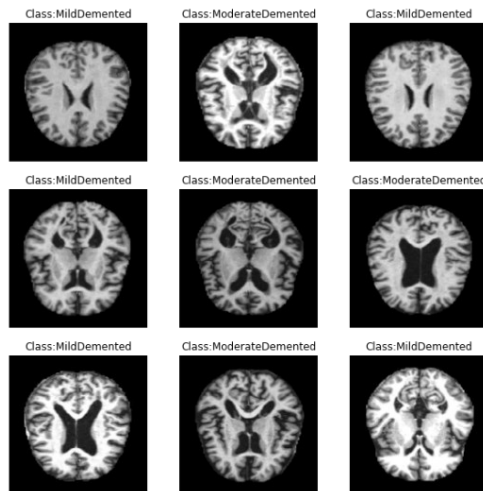


Gambar 6 Metodologi Penelitian



## Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini berjenis data sekunder yang bersumber dari Sarvesh Dubey. Dataset ini memiliki license *Open Data Commons Open Database License (ODbL) v1.0* yang dapat ditemukan pada situs *Kaggle*. Dataset terdiri dari gambar *Magnetic Resonance Imaging (MRI)*. Data ini memiliki empat kelas gambar yang dibagi menjadi data pelatihan dan data pengujian yaitu *Mild Demented*, *Moderate Demented*, *Non Demented*, *Very Mild Demented*. Total dataset yang digunakan sejumlah 6400 gambar dengan struktur data seperti terlihat pada Gambar 2. Dalam prakteknya kita membutuhkan fitur-fitur penting untuk mengoptimalkan kinerja tugas analisis data [23]. Kumpulan data yang tersedia dapat tercemar oleh faktor penghasil kesalahan lainnya seperti ketidakkonsistenan data dan data yang tidak lengkap. Tahap *preprocessing* data diperlukan untuk mengatasi masalah ini [24].



Gambar 7 Dataset alzheimer

## Pemrosesan Data

Pemrosesan data merupakan langkah penting sebelum data benar-benar diterapkan pada pembelajaran mesin (Mangkunegara and Purwono 2022). Langkah ini digunakan untuk memastikan data memiliki kualitas yang baik saat digunakan untuk melatih model (Nyirandayisabye et al. 2022). Teknik *image data generator* digunakan dalam pemrosesan dataset AD. Teknik ini digunakan untuk mengatasi ukuran dataset yang kecil dan menghasilkan sepuluh ribu gambar tambahan (Ahmad et al. 2022). Gambar input diproses dengan mengubahnya menjadi *tensor floating-point* ke dalam CNN. Tahapan yang digunakan dalam pemrosesan ini yaitu (1) membaca file gambar yang tersimpan dalam folder, (2) melakukan *decode* konten JPEG menjadi kisi piksel RGB dengan channels, (3) mengubah menjadi *tensor floating-point* sebagai data *input* ke CNN, (4) mengubah skala nilai *pixel* (antara 0 dan 255) ke *interval* [0,1].

Dataset AD yang digunakan rupanya memiliki kelas yang tidak seimbang (*imbalanced*) dan di dunia medis ini menjadi masalah umum dalam pengklasifikasian berbasis pembelajaran mendalam (Tartaglione et al. 2020) (Islam et al. 2021). Ketidakseimbangan kelas terjadi ketika beberapa kelas memiliki lebih banyak sampel secara signifikan dalam dataset pelatihan daripada yang lain. Telah terbukti bahwa ketidakseimbangan kelas dapat berdampak negatif pada pelatihan model CNN (Chamseddine et al. 2022). Solusi untuk mengatasi hal tersebut dapat dilakukan dengan menggunakan teknik *oversampling* yang telah banyak digunakan dalam pembelajaran mendalam serta telah terbukti

dapat diandalkan. Teknik *oversampling* SMOTE adalah teknik *oversampling* yang lebih canggih yang berupaya menyelesaikan masalah ini. Ini menghasilkan sampel sintetik baru dengan menginterpolasi data terdekat (Chamseddine et al. 2022). Di dalam penelitian ini, 80% dari data *preproceesing* digunakan untuk pelatihan, sedangkan 20% adalah digunakan untuk pengujian dan validasi model.

### Evaluation Metriks

Evaluasi metrik yang digunakan dalam klasifikasi CNN ini menggunakan metode standar. Evaluasi model yang digunakan yaitu akurasi, *precision*, *recall*, *f1-score* serta area dari *receiver operating characteristic (ROC) curve* and *the area under a ROC Curve (AUC)* (Deshpande, Estrela, and Patavardhan 2021). Akurasi ialah kapasitas sistem untuk menentukan jenis AD dengan benar. Rumus yang digunakan untuk menghitung akurasi sebagai berikut.

Table 1 Confusion Matrix

Kelas	Diklasifikasikan sebagai Positif	Diklasifikasikan sebagai Negatif
+	True Positive (TP)	False Positive (FP)
-	True Negative (TN)	False Negative (FN)

Tabel 1 adalah klasifikasi dari *confusion matrix*. *True positive (TP)* menunjukkan bahwa model klasifikasi dengan benar memberi label jumlah

tupel positif. *True Negative* (TN) menunjukkan bahwa model klasifikasi dengan benar memberi label jumlah tupel negatif. *False positive* (FP) menunjukkan bahwa model klasifikasi memberi label yang salah untuk jumlah tupel negatif. *False Negative* (FN) menunjukkan bahwa model klasifikasi memberi label yang salah untuk jumlah tupel positif. Akurasi adalah ukuran kinerja model klasifikasi dan merupakan persentase jumlah data yang diprediksi dengan benar dari total data (Katoch, Singh, and Tiwary 2022). Persamaan untuk menghitung ketelitian dapat dilihat pada Persamaan 14.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (14)$$

Pengukuran *performance metrics* terdiri dari presisi, *recall* dan *f1-score* (Xiong et al. 2022). Presisi adalah rasio positif atau derajat keandalan, yaitu proporsi prediksi berlabel positif yang benar terhadap prediksi positif keseluruhan (Luque et al. 2019). Persamaan untuk menghitung presisi dapat dilihat pada Persamaan 15.

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

*Recall* juga dikenal sebagai *true positive rate* atau sensitivitas. *Recall* juga disebut sebagai derajat keandalan model dalam mendeteksi data berlabel positif dengan benar (Luque et al. 2019). Persamaan untuk menghitung *recall* dapat dilihat pada persamaan 16.

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

*F1-score* merangkum semua hasil perhitungan *precision* dan *recall* dengan membuat rata-rata harmonik [27]. Persamaan untuk menghitung *F1-score* dapat dilihat pada persamaan 17.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (17)$$

Dalam penelitian ini kami melakukan komparasi dari beberapa jenis CNN architecture diantaranya ialah Inception V3, Inception Resnet, VGG16, Resnet50 dan Xception yang masing-masing dilatih dengan 100 epoch. Kami juga terinspirasi dari kode program yang sudah dikerjakan oleh (Venkatesh et al.).

## BAB V

# Hasil Model Klasifikasi CNN

### Model Pembelajaran

Eksperimen dilakukan untuk memprediksi penyakit alzheimer dengan menerapkan berbagai model transfer learning pada CNN. Tahap ini dilakukan dengan mengimport modul-modul yang harus diinstal agar model dapat berjalan.

### Import Pustaka

Kode program berikut merupakan pengambilan pustaka model transfer learning yang digunakan pada web Kaggle.

#### Kode. 1 Import Pustaka Transfer Learning

```
1. from
   tensorflow.keras.applications.inception_v3
   import InceptionV3
2. from tensorflow.keras.applications import
   ResNet50
3. from tensorflow.keras.applications.xception
   import Xception
4. from keras.applications.vgg16 import VGG16
5. from tensorflow.keras.applications import
   InceptionResNetV2
```

Tahap selanjutnya ialah melakukan pemrosesan data. Dataset diproses dengan menggunakan pustaka *Image Generator*.

### Pemrosesan Dataset

#### Kode. 2 Pemrosesan Dataset

```
1. from tensorflow.keras.preprocessing.image
   import ImageDataGenerator as IDG
```

*Image Generator* dijadikan data sample untuk memproses dataset yang tersedia. *Generator* akan menjalankan data gambar serta menerapkan transformasi acak ke setiap gambar individu saat diteruskan ke model sehingga tidak dapat melihat gambar yang sama dua kali selama pelatihan. Transformasi ini adalah parameter pada generator yang dapat diatur saat dipakai dan dapat mencakup rotasi, geser, membalik, dan memperbesar. Kode program berikut merupakan penggunaan pustaka *Image Generator*.

**Kode. 3** Proses Pengambilan Direktori Gambar

```
1. base_dir = "/kaggle/input/alzheimers-
   dataset-4-class-of-images/Alzheimer_s
   Dataset/"
2. root_dir = "./"
3. test_dir = base_dir + "test/"
4. train_dir = base_dir + "train/"
5. work_dir = root_dir + "dataset/"
6.
7. if os.path.exists(work_dir):
8.     remove_tree(work_dir)
9.
10.
11. os.mkdir(work_dir)
12. copy_tree(train_dir, work_dir)
13. copy_tree(test_dir, work_dir)
14. print("Working Directory Contents:",
        os.listdir(work_dir))
```

**Kode. 4** Proses Pembuatan Kelas dari Dataset

```
1. WORK_DIR = './dataset/'
2.
```

```

3. CLASSES = [ 'NonDemented',
4.             'VeryMildDemented',
5.             'MildDemented',
6.             'ModerateDemented' ]
7.
8. IMG_SIZE = 176
9. IMAGE_SIZE = [176, 176]
10. DIM = (IMG_SIZE, IMG_SIZE)

```

#### **Kode. 5** Proses Penggunaan Library Image Generator

```

1. ZOOM = [.99, 1.01]
2. BRIGHT_RANGE = [0.8, 1.2]
3. HORZ_FLIP = True
4. FILL_MODE = "constant"
5. DATA_FORMAT = "channels_last"
6.
7. work_dr = IDG(rescale = 1./255,
  brightness_range=BRIGHT_RANGE,
  zoom_range=ZOOM, data_format=DATA_FORMAT,
  fill_mode=FILL_MODE,
  horizontal_flip=HORZ_FLIP)
8.
9. train_data_gen =
  work_dr.flow_from_directory(directory=WORK_
  DIR, target_size=DIM, batch_size=6500,
  shuffle=False)

```

#### **Kode. 6** Proses Mencetak Gambar Berdasarkan Kelas

```

1. def show_images(generator,y_pred=None):
2.
3.     # get image lables
4.     labels =dict(zip([0,1,2,3], CLASSES))
5.
6.     # get a batch of images
7.     x,y = generator.next()
8.
9.     # display a grid of 9 images
10.    plt.figure(figsize=(10, 10))
11.    if y_pred is None:
12.        for i in range(9):
13.            ax = plt.subplot(3, 3, i + 1)

```



```

14.         idx = randint(0, 6400)
15.         plt.imshow(x[idx])
16.         plt.axis("off")
17.
18.         plt.title("Class:{}".format(labels[np.argmax(
19.             x(y[idx]))]))
20.     else:
21.         for i in range(9):
22.             ax = plt.subplot(3, 3, i + 1)
23.             plt.imshow(x[i])
24.             plt.axis("off")
25.             plt.title("Actual:{}".format(
26.                 labels[np.argmax(y[i
27.                     ])],labels[y_pred[i]]))
28.
29. # Display Train Images
30. show_images(train_data_gen)

```

## Mengambil data ImageDataGenerator

Setelah pustaka dijalankan, data gambar selanjutnya di ambil dari iterator *image generator*. Kita juga dapat mengambil dimensi dari dataset tersebut.

### **Kode. 7** Proses Pengambilan Data dan Dimensi Dataset

```

1. train_data, train_labels =
   train_data_gen.next()

2. print(train_data.shape, train_labels.shape)

```

Klasifikasi yang tidak seimbang melibatkan pengembangan model prediktif pada kumpulan data klasifikasi yang memiliki ketidakseimbangan kelas yang parah. Tantangan dalam bekerja dengan kumpulan data yang tidak seimbang adalah bahwa sebagian besar teknik pembelajaran mesin akan diabaikan, dan pada gilirannya memiliki kinerja yang

buruk pada kelas minoritas, meskipun biasanya kinerja pada kelas minoritaslah yang paling penting. Salah satu pendekatan untuk mengatasi kumpulan data yang tidak seimbang adalah dengan mengambil sampel kelas minoritas secara berlebihan. Pendekatan paling sederhana melibatkan duplikasi contoh di kelas minoritas, meskipun contoh ini tidak menambahkan informasi baru ke model. Sebaliknya, contoh baru dapat disintesis dari contoh yang sudah ada. Ini adalah jenis augmentasi data untuk kelas minoritas dan disebut sebagai Teknik *Oversampling Minoritas Sintetis*, atau disingkat SMOTE (*Synthetic Minority Oversampling Technique*). Berikut adalah kode program untuk melakukan teknik SMOTE pada dataset alzheimer yang tidak seimbang.

**Kode. 8** Proses SMOTE pada data tidak seimbang

```
1. sm = SMOTE(random_state=42)
2.
3. train_data, train_labels =
   sm.fit_resample(train_data.reshape(-1,
   IMG_SIZE * IMG_SIZE * 3), train_labels)
4.
5. train_data = train_data.reshape(-1,
   IMG_SIZE, IMG_SIZE, 3)
6.
7. print(train_data.shape, train_labels.shape)
```

Tahap selanjutnya, kita dapat melakukan pembagian data menjadi data latih dan data uji. Komposisi yang diterapkan ialah 80% : 20%. Berikut adalah kode program untuk melakukan splitting dataset alzheimer.

**Kode. 9** Proses Splitting Dataset

```

1. train_data, test_data, train_labels,
   test_labels = train_test_split(train_data,
   train_labels, test_size = 0.2,
   random_state=42)
2. train_data, val_data, train_labels,
   val_labels = train_test_split(train_data,
   train_labels, test_size = 0.2,
   random_state=42)

```

## Pembuatan Model CNN

Berdasarkan metode penelitian kita akan membuat lima model *transfer learning* yang akan dibandingkan. Model tersebut antara lain InceptionV3, Inception ResNet, VGG16, ResNet50 dan Xception. Berikut adalah kode program untuk 5 model yang dibuat.

## Model InceptionV3

### Kode. 10 Model InceptionV3

```

1. inceptionv3_model =
   InceptionV3(input_shape=(176, 176, 3),
   include_top=False, weights="imagenet")
2.
3. for layer in inceptionv3_model.layers:
4.     layer.trainable=False
5.
6. custom_inceptionv3_model = Sequential([
7.     inceptionv3_model,
8.     Dropout(0.5),
9.     GlobalAveragePooling2D(),
10.    Flatten(),
11.    BatchNormalization(),
12.    Dense(512, activation='relu'),
13.    BatchNormalization(),
14.    Dropout(0.5),
15.    Dense(256, activation='relu'),

```

```

16.         BatchNormalization(),
17.         Dropout(0.5),
18.         Dense(128, activation='relu'),
19.         BatchNormalization(),
20.         Dropout(0.5),
21.         Dense(64, activation='relu'),
22.         Dropout(0.5),
23.         BatchNormalization(),
24.         Dense(4, activation='softmax')
25.     ], name = "inception_cnn_model")

```

26.

```

27. class
    MyCallback(tf.keras.callbacks.Callback):
28.     def on_epoch_end(self, epoch, logs={}):
29.         if logs.get('acc') > 0.99:
30.             print("\nReached accuracy
                threshold! Terminating training.")
31.             self.model.stop_training = True
32.
33. my_callback = MyCallback()
34.
35. rop_callback =
    ReduceLROnPlateau(monitor="val_loss",
                    patience=3)

```

36.

```

37. METRICS =
    [tf.keras.metrics.CategoricalAccuracy(name='
        acc'),
38.         tf.keras.metrics.AUC(name='auc'),
39.         tfa.metrics.F1Score(num_classes=4)]
40.
41. CALLBACKS = [my_callback, rop_callback]
42.
43. custom_inceptionv3_model.compile(optimizer='
        rmsprop',
44.         loss=tf.losses.CategoricalCrossentropy(),
45.         metrics=METRICS)
46.

```

```

47. custom_inceptionv3_model.summary()
48.
49. EPOCHS = 100
50.
51. history =
    custom_inception_model.fit(train_data,
        train_labels, validation_data=(val_data,
            val_labels), callbacks=CALLBACKS,
            epochs=EPOCHS)

```

## Model Inception Resnet

### Kode. 11 Model Inception Resnet

```

1. inception_resnet_v2 = InceptionResNetV2(
2.     include_top=False,
3.     weights="imagenet",
4.     input_shape=None)
5.
6. for layer in inception_resnet_v2.layers:
7.     layer.trainable=False
8.
9. custom_inceptionresnet_model = Sequential([
10.     inception_resnet_v2,
11.     Dropout(0.5),
12.     GlobalAveragePooling2D(),
13.     Flatten(),
14.     BatchNormalization(),
15.     Dense(512, activation='relu'),
16.     BatchNormalization(),
17.     Dropout(0.5),
18.     Dense(256, activation='relu'),
19.     BatchNormalization(),
20.     Dropout(0.5),
21.     Dense(128, activation='relu'),
22.     BatchNormalization(),
23.     Dropout(0.5),
24.     Dense(64, activation='relu'),
25.     Dropout(0.5),
26.     BatchNormalization(),

```

```

27.         Dense(4, activation='softmax')
28.     ], name = "inception_cnn_model")
29.
30. class
    MyCallback(tf.keras.callbacks.Callback):
31.     def on_epoch_end(self, epoch, logs={}):
32.         if logs.get('acc') > 0.99:
33.             print("\nReached accuracy
                threshold! Terminating training.")
34.             self.model.stop_training = True
35.
36. my_callback = MyCallback()
37.
38. rop_callback =
    ReduceLROnPlateau(monitor="val_loss",
                    patience=3)
39.
40. METRICS =
    [tf.keras.metrics.CategoricalAccuracy(name='
        acc'),
41.     tf.keras.metrics.AUC(name='auc'),
42.     tf.keras.metrics.F1Score(num_classes=4)]
43.
44. CALLBACKS = [my_callback, rop_callback]
45.
46. custom_inceptionresnet_model.compile(optimizer='rmsprop',
47.                                     loss=tf.losses.CategoricalCrossentropy(),
48.                                     metrics=METRICS)
49.
50. custom_inceptionresnet_model.summary()
51.
52. EPOCHS = 100
53.
54. history =
    custom_inceptionresnet_model.fit(train_data,
                                    train_labels, validation_data=(val_data,

```

```
val_labels), callbacks=CALLBACKS,  
epochs=EPOCHS)
```

## Model VGG16

### Kode. 12 Model VGG 16

```
1. vgg_model =  
   VGG16(weights='imagenet',include_top=False)  
2.  
3. for layer in vgg_model.layers:  
4.     layer.trainable=False  
5.  
6. custom_vggmodel = Sequential([  
7.     vgg_model,  
8.     Dropout(0.5),  
9.     GlobalAveragePooling2D(),  
10.    Flatten(),  
11.    BatchNormalization(),  
12.    Dense(512, activation='relu'),  
13.    BatchNormalization(),  
14.    Dropout(0.5),  
15.    Dense(256, activation='relu'),  
16.    BatchNormalization(),  
17.    Dropout(0.5),  
18.    Dense(128, activation='relu'),  
19.    BatchNormalization(),  
20.    Dropout(0.5),  
21.    Dense(64, activation='relu'),  
22.    Dropout(0.5),  
23.    BatchNormalization(),  
24.    Dense(4, activation='softmax')  
25. ], name = "inception_cnn_model")  
26.  
27. class  
    MyCallback(tf.keras.callbacks.Callback):  
28.     def on_epoch_end(self, epoch, logs={}):  
29.         if logs.get('acc') > 0.99:  
30.             print("\nReached accuracy  
    threshold! Terminating training.")
```

```
31.         self.model.stop_training = True
32.
33. my_callback = MyCallback()
34.
35. rop_callback =
    ReduceLROnPlateau(monitor="val_loss",
        patience=3)
```

```
36.
```

```
37. METRICS =
    [tf.keras.metrics.CategoricalAccuracy(name='
    acc'),
38.         tf.keras.metrics.AUC(name='auc'),
39.         tfa.metrics.F1Score(num_classes=4)]
40.
41. CALLBACKS = [my_callback, rop_callback]
42.
43. custom_vggmodel.compile(optimizer='rmsprop',
44.         loss=tf.losses.CategoricalCrossentropy(),
45.         metrics=METRICS)
46.
47. custom_vggmodel.summary()
```

```
48.
```

```
49. EPOCHS = 100
```

```
50.
```

```
51. history = custom_vggmodel.fit(train_data,
    train_labels, validation_data=(val_data,
    val_labels), callbacks=CALLBACKS,
    epochs=EPOCHS)
```

## Model Resnet50

### Kode. 13 Model Resnet50

```
1. resnet_model = ResNet50(
2.     include_top=False,
3.     weights="imagenet",
4.     input_tensor=None,
5.     input_shape=None,
```



6.	pooling=None,
7.	classes=1000
8.	)
9.	
10.	for layer in resnet_model.layers:
11.	layer.trainable=False
12.	
13.	custom_resnet_model = Sequential([
14.	resnet_model,
15.	Dropout(0.5),
16.	GlobalAveragePooling2D(),
17.	Flatten(),
18.	BatchNormalization(),
19.	Dense(512, activation='relu'),
20.	BatchNormalization(),
21.	Dropout(0.5),
22.	Dense(256, activation='relu'),
23.	BatchNormalization(),
24.	Dropout(0.5),
25.	Dense(128, activation='relu'),
26.	BatchNormalization(),
27.	Dropout(0.5),
28.	Dense(64, activation='relu'),
29.	Dropout(0.5),
30.	BatchNormalization(),
31.	Dense(4, activation='softmax')
32.	], name = "inception_cnn_model")
33.	
34.	class
	MyCallback(tf.keras.callbacks.Callback):
35.	def on_epoch_end(self, epoch, logs={}):
36.	if logs.get('acc') > 0.99:
37.	print("\nReached accuracy
	threshold! Terminating training.")
38.	self.model.stop_training = True
39.	
40.	my_callback = MyCallback()
41.	

```

42. rop_callback =
    ReduceLROnPlateau(monitor="val_loss",
    patience=3)
43.
44. METRICS =
    [tf.keras.metrics.CategoricalAccuracy(name='
    acc'),
45.         tf.keras.metrics.AUC(name='auc'),
46.         tfa.metrics.F1Score(num_classes=4)]
47.
48. CALLBACKS = [my_callback, rop_callback]
49.
50. custom_resnet_model.compile(optimizer='rmspr
    op',
51.         loss=tf.losses.CategoricalCrossentropy(),
52.         metrics=METRICS)
53.
54. custom_resnet_model.summary()
55.
56. EPOCHS = 100
57.
58. history =
    custom_resnet_model.fit(train_data,
    train_labels, validation_data=(val_data,
    val_labels), callbacks=CALLBACKS,
    epochs=EPOCHS)

```

## Model Xception

### Kode. 14 Model Xception

```

1. xception_model = Xception(
2.     include_top=False,
3.     weights="imagenet",
4.     input_tensor=None,
5.     input_shape=None,
6.     pooling=None,
7.     classes=1000,

```

8. classifier_activation="softmax",
9. )
10.
11. for layer in xception_model.layers:
12. layer.trainable=False
13.
14. custom_xception_model = Sequential([
15. xception_model,
16. Dropout(0.5),
17. GlobalAveragePooling2D(),
18. Flatten(),
19. BatchNormalization(),
20. Dense(512, activation='relu'),
21. BatchNormalization(),
22. Dropout(0.5),
23. Dense(256, activation='relu'),
24. BatchNormalization(),
25. Dropout(0.5),
26. Dense(128, activation='relu'),
27. BatchNormalization(),
28. Dropout(0.5),
29. Dense(64, activation='relu'),
30. Dropout(0.5),
31. BatchNormalization(),
32. Dense(4, activation='softmax')
33. ], name = "inception_cnn_model")
34.
35. class
MyCallback(tf.keras.callbacks.Callback):
36. def on_epoch_end(self, epoch, logs={}):
37. if logs.get('acc') > 0.99:
38. print("\nReached accuracy
threshold! Terminating training.")
39. self.model.stop_training = True
40.
41. my_callback = MyCallback()
42.
43. rop_callback =
ReduceLROnPlateau(monitor="val_loss",
patience=3)

```

44.
45. METRICS =
    [tf.keras.metrics.CategoricalAccuracy(name='
    acc'),
46.         tf.keras.metrics.AUC(name='auc'),
47.         tfa.metrics.F1Score(num_classes=4)]
48.
49. CALLBACKS = [my_callback, rop_callback]
50.
51. custom_xception_model.compile(optimizer='rms
    prop',
52.         loss=tf.losses.CategoricalCrossentropy(),
53.         metrics=METRICS)
54.
55. custom_xception_model.summary()
56.
57. EPOCHS = 100
58.
59. history =
    custom_resnet_model.fit(train_data,
        train_labels, validation_data=(val_data,
        val_labels), callbacks=CALLBACKS,
        epochs=EPOCHS)

```

Dari hasil percobaan pelatihan dengan menggunakan 100 *epoch* didapatkan hasil grafik tren metrik selama pelatihan yang dapat dilihat pada Gambar 8. Kode setiap model *transfer learning* dapat dilihat pada kode program berikut.

#### **Kode. 15** Trend Metriks Training

```

1. fig, ax = plt.subplots(1, 3, figsize = (30,
    5))
2. ax = ax.ravel()
3.

```

```

4. for i, metric in enumerate(["acc", "auc",
    "loss"]):
5.     ax[i].plot(history.history[metric])
6.     ax[i].plot(history.history["val_" +
    metric])
7.     ax[i].set_title("Model
    {}".format(metric))
8.     ax[i].set_xlabel("Epochs")
9.     ax[i].set_ylabel(metric)
10.    ax[i].legend(["train", "val"])

```

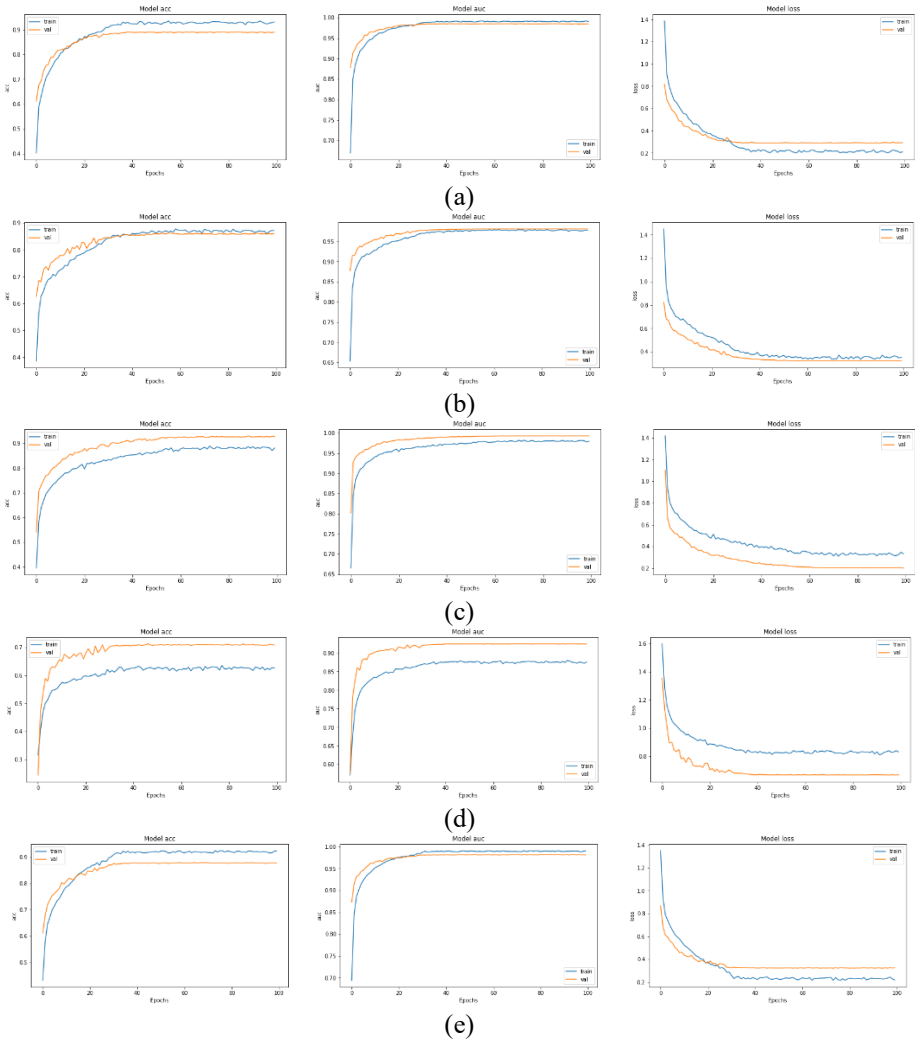
Terlihat nilai AUC dari masing-masing model berada pada titik atas kurva yang menandakan bahwa proses pengujian model untuk penyakit alzheimer ini dikategorikan sebagai model yang baik dalam melakukan klasifikasi. Pada proses ini kita mendapatkan nilai akurasi dari pelatihan untuk setiap jenis model transfer learning yang dapat dilihat pada Tabel 2.

*Table 2 Hasil Pembelajaran Model*

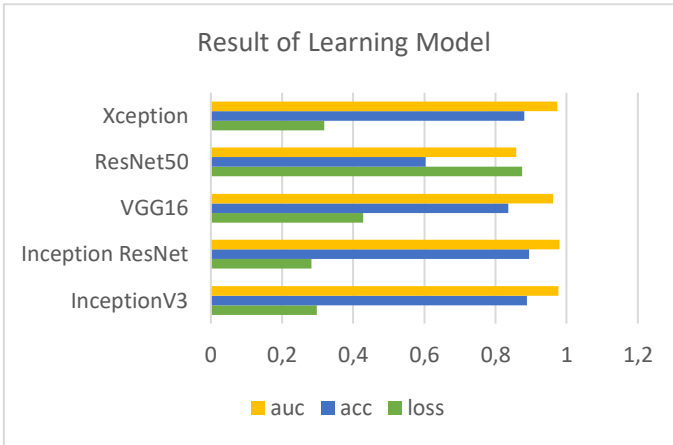
<b>Model</b>	<b>loss</b>	<b>acc</b>	<b>auc</b>
InceptionV3	0,297	0,889	0,977
Inception ResNet	0,283	0,894	0,980
VGG16	0,428	0,836	0,962
ResNet50	0,875	0,603	0,858
Xception	0,318	0,881	0,974

Berdasarkan Tabel 2, kita dapat melihat bahwa nilai loss tertinggi dikeluarkan oleh model ResNet50 sebesar 0,875 dan nilai loss terendah dikeluarkan oleh model Inception ResNet sebesar 0,283. nilai akurasi pada tahap pengujian, model Inception ResNet memiliki nilai akurasi tertinggi sebesar 0,894 dan model ResNet50 memiliki nilai akurasi terendah sebesar 0,603. Sedangkan nilai AUC

tertinggi diperoleh model InceptionV3 sebesar 0,977 dan nilai AUC terendah diperoleh model ResNet50 sebesar 0,858. Ilustrasi proses penguujian model dapat dilihat pada Gambar 9.



Gambar 8 Tren metrik selama pelatihan



Gambar 9 Hasil Komparasi Transfer Learning

## Model Pengujian

Dari sisi data pengujian, kita telah berhasil mendapatkan data accuracy, precission, recall dan fl-score dari masing-masing kelas penyakit alzheimer. Adapun hasil pengujian dari masing-masing model transfer learning CNN dapat dilihat pada Tabel 3. Nilai-nilai tersebut dihasilkan dari perhitungan rumus evaluation metrik pada jenis pembelajaran klasifikasi algoritma CNN. Adapun kode program untuk melakukan evaluasi model sebagai berikut.

<b>Kode. 16</b> Evaluasi Model
1. <code>print("Testing Accuracy: %.2f%%"%(test_scores[1] * 100))</code>
2.
3. <code>pred_labels =     <b>custom_machine_model</b>.predict(test_data)</code>
4.
5. <code>def roundoff(arr):</code>
6. <code>arr[np.argmax(arr != arr.max())] = 0</code>
7. <code>arr[np.argmax(arr == arr.max())] = 1</code>
8. <code>return arr</code>

```

9.
10. for labels in pred_labels:
11.     labels = roundoff(labels)
12.
13. print(classification_report(test_labels,
    pred_labels, target_names=CLASSES))
14.
15. pred_ls = np.argmax(pred_labels, axis=1)
16. test_ls = np.argmax(test_labels, axis=1)
17.
18. conf_arr = confusion_matrix(test_ls,
    pred_ls)
19.
20. plt.figure(figsize=(8, 6), dpi=80,
    facecolor='w', edgecolor='k')
21.
22. ax = sns.heatmap(conf_arr, cmap='Greens',
    annot=True, fmt='d', xticklabels= CLASSES,
23.                 yticklabels=CLASSES)
24.
25. plt.title('Alzheimer\'s Disease Diagnosis')
26. plt.xlabel('Prediction')
27. plt.ylabel('Truth')
28. plt.show(ax)

```

*Table 3 Hasil Evaluasi Model*

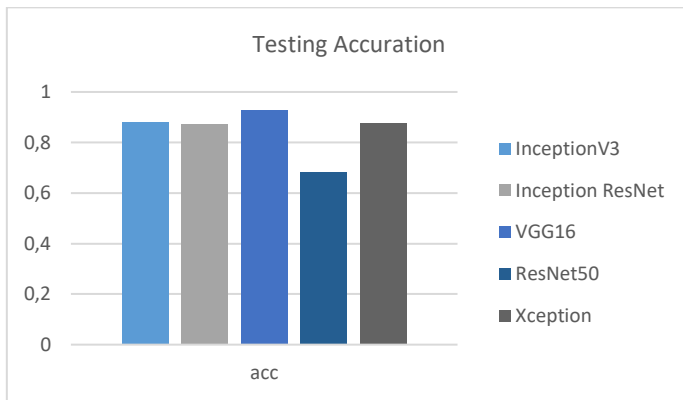
Evaluation	Class	InceptionV3	Inception ResNet	VGG16	ResNet50	Xception
Precision	1	0.89	0.89	0.95	0.70	0.90
	2	1.00	1.00	1.00	0.97	1.00
	3	0.84	0.84	0.88	0.71	0.83
	4	0.75	0.75	0.88	0.42	0.78
Recall	1	0.95	0.95	0.97	0.66	0.94
	2	1.00	1.00	1.00	1.00	1.00
	3	0.76	0.76	0.88	0.50	0.79
	4	0.77	0.77	0.86	0.56	0.79
F1-Score	1	0.92	0.92	0.96	0.68	0.92
	2	1.00	1.00	1.00	0.99	1.00
	3	0.80	0.80	0.88	0.59	0.81
	4	0.76	0.76	0.87	0.48	0.79



Akurasi dari masing-masing model yang dihasilkan pada tahap pengujian ini disajikan pada Tabel 4. Nilai akurasi tersebut selanjutnya dapat dibandingkan kinerjanya dengan cara melihat model dengan akurasi tertinggi. Adapun ilustrasi perbandingan model ini dapat dilihat pada Gambar 10.

*Table 4 Hasil Model Pelatihan*

<b>Model</b>	<b>acc</b>
InceptionV3	0,882
Inception ResNet	0,871
VGG16	0,927
ResNet50	0,682
Xception	0,878



*Gambar 10 Hasil Akurasi Pengujian Model*

Berdasarkan Tabel 3 dan Gambar 4, terlihat bahwa model VGG16 menempati urutan nilai akurasi teratas dengan nilai sebesar 0,927, sedangkan untuk nilai akurasi terbawah dimiliki oleh model ResNet50. Jika dibandingkan dengan penelitian terkait, model VGG16 ini dalam melakukan klasifikasi penyakit Alzheimer masih lebih rendah nilai akurasinya

dibandingkan dengan penelitian Kruthika dan Geenedy, namun masih lebih tinggi dibandingkan dengan penelitian yang dilakukan oleh Liu dan Bringas.

## PENUTUP

Berdasarkan hasil eksperimen yang telah dilakukan, penelitian ini telah berhasil melakukan komparasi model transfer learning CNN untuk penyakit alzheimer. Dataset telah melewati tahap pemrosesan data sehingga bisa digunakan untuk membuat model pembelajaran mendalam dengan CNN. Pada tahap pembelajaran, terlihat nilai AUC dari masing-masing model diatas 0,85 yang artinya model telah belajar dengan baik untuk melakukan klasifikasi penyakit alzheimer. Nilai loss tertinggi dikeluarkan oleh model ResNet50 sebesar 0,875 dan nilai loss terendah dikeluarkan oleh model Inception ResNet sebesar 0,283. nilai akurasi pada tahap pengujian, model Inception ResNet memiliki nilai akurasi tertinggi sebesar 0,894 dan model ResNet50 memiliki nilai akurasi terendah sebesar 0,603. Sedangkan nilai AUC tertinggi diperoleh model InceptionV3 sebesar 0,977 dan nilai AUC terendah diperoleh model ResNet50 sebesar 0,858. Pada tahap pengujian model, terlihat bahwa model VGG16 menempati urutan nilai akurasi teratas dengan nilai sebesar 0,927, sedangkan untuk nilai akurasi terbawah dimiliki oleh model ResNet50.

## DAFTAR PUSTAKA

- Ahmad, Muhammad, Saima Sadiq, Ala' Abdulmajid Eshmawi, Ala Saleh Alluhaidan, Muhammad Umer, Saleem Ullah, and Michele Nappi. 2022. "Industry 4.0 Technologies and Their Applications in Fighting COVID-19 Pandemic Using Deep Learning Techniques." *Computers in Biology and Medicine* 145:105418. doi: <https://doi.org/10.1016/j.compbiod.2022.105418>.
- Alnussairi, Muqdad Hanoon Dawood, and Abdullahi Abdu Ibrahim. 2022. "Malaria Parasite Detection Using Deep Learning Algorithms Based on (CNNs) Technique." *Computers and Electrical Engineering* 103:108316. doi: <https://doi.org/10.1016/j.compeleceng.2022.108316>.
- Alorf, Abdulaziz, and Muhammad Usman Ghani Khan. 2022. "Multi-Label Classification of Alzheimer's Disease Stages from Resting-State FMRI-Based Correlation Connectivity Data and Deep Learning." *Computers in Biology and Medicine* 151:106240. doi: <https://doi.org/10.1016/j.compbiod.2022.106240>.
- Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaria, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. 2021. *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*. Vol. 8. Springer International Publishing.

Aradhya, Abhay M. S., Andri Ashfahani, Fienny Angelina, Mahardhika Pratama, Rodrigo Fernandes de Mello, and Suresh Sundaram. 2022. "Autonomous CNN (AutoCNN): A Data-Driven Approach to Network Architecture Determination." *Information Sciences* 607:638–53. doi: <https://doi.org/10.1016/j.ins.2022.05.100>.

Bezdan, Timea, and Nebojša Bačanin Džakula. 2019. "Convolutional Neural Network Layers and Architectures." Pp. 445–51 in *International Scientific Conference on Information Technology and Data Related Research*.

Bhatt, Dulari, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. 2021. "Cnn Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope." *Electronics (Switzerland)* 10(20):1–28. doi: [10.3390/electronics10202470](https://doi.org/10.3390/electronics10202470).

Bringas, Santos, Sergio Salomón, Rafael Duque, Carmen Lage, and José Luis Montaña. 2020. "Alzheimer's Disease Stage Identification Using Deep Learning Models." *Journal of Biomedical Informatics* 109:103514. doi: <https://doi.org/10.1016/j.jbi.2020.103514>.

Cao, Jianfang, Minmin Yan, Yiming Jia, Xiaodong Tian, and Zibang Zhang. 2021. "Application of a Modified Inception-v3 Model in the Dynasty-Based Classification of Ancient Murals." *Eurasip Journal on Advances in Signal Processing* 2021(1). doi: [10.1186/s13634-021-](https://doi.org/10.1186/s13634-021-)

00740-8.

Chamseddine, Ekram, Nesrine Mansouri, Makram Soui, and Mourad Abed. 2022. "Handling Class Imbalance in COVID-19 Chest X-Ray Images Classification: Using SMOTE and Weighted Loss." *Applied Soft Computing* 129:109588. doi: <https://doi.org/10.1016/j.asoc.2022.109588>.

Chandler, Hannah, Richard Wise, David Linden, Julie Williams, Kevin Murphy, and Thomas Matthew Lancaster. 2022. "Alzheimer's Genetic Risk Effects on Cerebral Blood Flow across the Lifespan Are Proximal to Gene Expression." *Neurobiology of Aging* 120:1–9. doi: <https://doi.org/10.1016/j.neurobiolaging.2022.08.001>.

Chen, Haiyan, Yu Yang, and Suning Zhang. 2020. "Learning Robust Scene Classification Model with Data Augmentation Based on Xception." *Journal of Physics: Conference Series* 1575(1):12009. doi: 10.1088/1742-6596/1575/1/012009.

Coulibaly, Solemane, Bernard Kamsu-Foguem, Dantouma Kamissoko, and Daouda Traore. 2022. "Deep Learning for Precision Agriculture: A Bibliometric Analysis." *Intelligent Systems with Applications* 16:200102. doi: <https://doi.org/10.1016/j.iswa.2022.200102>.

Coursera. 2022. "Deep Learning vs. Machine Learning: Beginner's Guide." *Coursera*.

Retrieved (<https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide>).

Dambal, Sharada Suresh, Mahesh Kumar Doddananjedevuru, and Sudarshan Budnur Gopalakrishna. 2022. "Premature Ventricular Contraction Classification Based on Spiral Search - Manta Ray Foraging and Bi-LSTM." *International Journal of Intelligent Engineering and Systems* 15(6):1–10. doi: 10.22266/ijies2022.1231.01.

Deshpande, Anand, Vania V Estrela, and Prashant Patavardhan. 2021. "The DCT-CNN-ResNet50 Architecture to Classify Brain Tumors with Super-Resolution, Convolutional Neural Network, and the ResNet50." *Neuroscience Informatics* 1(4):100013. doi: <https://doi.org/10.1016/j.neuri.2021.100013>.

Ding, Chen, Ying Li, Yong Xia, Lei Zhang, and Yanning Zhang. 2018. "Automatic Kernel Size Determination for Deep Neural Networks Based Hyperspectral Image Classification." *Remote Sensing* 10(3). doi: 10.3390/rs10030415.

Ebrahimighahnaveh, Mr Amir, Suhuai Luo, and Raymond Chiong. 2020. "Deep Learning to Detect Alzheimer's Disease from Neuroimaging: A Systematic Literature Review." *Computer Methods and Programs in Biomedicine* 187:105242. doi: <https://doi.org/10.1016/j.cmpb.2019.105242>.

Ein Shoka, Athar A., Mohamed M. Dessouky,

Ayman El-Sayed, and Ezz El-Din Hemdan. 2022. “An Efficient CNN Based Epileptic Seizures Detection Framework Using Encrypted EEG Signals for Secure Telemedicine Applications.” *Alexandria Engineering Journal*. doi: <https://doi.org/10.1016/j.aej.2022.10.014>.

EL-Geneedy, Marwa, Hossam El-Din Moustafa, Fahmi Khalifa, Hatem Khater, and Eman AbdElhalim. 2023. “An MRI-Based Deep Learning Approach for Accurate Detection of Alzheimer’s Disease.” *Alexandria Engineering Journal* 63:211–21. doi: <https://doi.org/10.1016/j.aej.2022.07.062>.

Fu, Guangtao, Yiwen Jin, Siao Sun, Zhiguo Yuan, and David Butler. 2022. “The Role of Deep Learning in Urban Water Management: A Critical Review.” *Water Research* 223:118973. doi: <https://doi.org/10.1016/j.watres.2022.118973>.

Ghosh, Anirudha, Abu Sufian, Farhana Sultana, Amlan Chakrabarti, and Debashis De. 2019. “Fundamental Concepts of Convolutional Neural Network.” Pp. 519–67 in *Intelligent Systems Reference Library*. Vol. 172.

Gonçalves, Caroline Barcelos, Jefferson R. Souza, and Henrique Fernandes. 2022. “CNN Architecture Optimization Using Bio-Inspired Algorithms for Breast Cancer Detection in Infrared Images.” *Computers in Biology and Medicine* 142:105205. doi: <https://doi.org/10.1016/j.compbiomed.2021.105205>.



He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition." Pp. 770–78 in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Islam, M. M., F. Karray, R. Alhadjj, and J. Zeng. 2021. "A Review on Deep Learning Techniques for the Diagnosis of Novel Coronavirus (COVID-19)." *IEEE Access* 9:30551–72. doi: 10.1109/ACCESS.2021.3058537.

Kale, Sandip A. 2019. "E-Commerce for Future & Trends Smart Integrated Approach for Future E-Commerce Jewelry Business Using 3D Printing." *E - Commerce for Future & Trends* 6(1):83–90.

Katoch, Shagun, Varsha Singh, and Uma Shanker Tiwary. 2022. "Indian Sign Language Recognition System Using SURF with SVM and CNN." *Array* 100141. doi: <https://doi.org/10.1016/j.array.2022.100141>.

Katsumata, Yuriko, Lincoln M. Shade, Timothy J. Hohman, Julie A. Schneider, David A. Bennett, Jose M. Farfel, Walter A. Kukull, David W. Fardo, and Peter T. Nelson. 2022. "Multiple Gene Variants Linked to Alzheimer's-Type Clinical Dementia via GWAS Are Also Associated with Non-Alzheimer's Neuropathologic Entities." *Neurobiology of Disease* 174:105880. doi: <https://doi.org/10.1016/j.nbd.2022.105880>.

Ke, Qihong, Jun Liu, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid.

2018. “Computer Vision for Human-Machine Interaction.” Pp. 127–45 in *Computer Vision For Assistive Healthcare*. Elsevier Ltd.

Khan, Afreen, and Swaleha Zubair. 2022.

“Development of a Three Tiered Cognitive Hybrid Machine Learning Algorithm for Effective Diagnosis of Alzheimer’s Disease.” *Journal of King Saud University - Computer and Information Sciences* 34(10, Part A):8000–8018. doi:  
<https://doi.org/10.1016/j.jksuci.2022.07.016>.

Kong, Lingzhi, and Jinyong Cheng. 2022.

“Classification and Detection of COVID-19 X-Ray Images Based on DenseNet and VGG16 Feature Fusion.” *Biomedical Signal Processing and Control* 77:103772. doi:  
<https://doi.org/10.1016/j.bspc.2022.103772>.

Kruthika, K. R., Rajeswari, and H. D. Maheshappa. 2019. “CBIR System Using Capsule Networks and 3D CNN for Alzheimer’s Disease Diagnosis.” *Informatics in Medicine Unlocked* 16:100227. doi:  
<https://doi.org/10.1016/j.imu.2019.100227>.

Lin, Manman, Haisheng Ou, Peng Zhang, Yanhong Meng, Shenghao Wang, Jing Chang, Aiguo Shen, and Jiming Hu. 2022. “Laser Tweezers Raman Spectroscopy Combined with Machine Learning for Diagnosis of Alzheimer’s Disease.” *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* 280:121542. doi:  
<https://doi.org/10.1016/j.saa.2022.121542>.

Liu, Manhua, Fan Li, Hao Yan, Kundong Wang, Yixin Ma, Li Shen, and Mingqing Xu. 2020. “A Multi-Model Deep Convolutional Neural Network for Automatic Hippocampus Segmentation and Classification in Alzheimer’s Disease.” *NeuroImage* 208:116459. doi: <https://doi.org/10.1016/j.neuroimage.2019.116459>.

Liu, Zhiyong, Chuan Yang, Jun Huang, Shaopeng Liu, Yumin Zhuo, and Xu Lu. 2021. “Deep Learning Framework Based on Integration of S-Mask R-CNN and Inception-v3 for Ultrasound Image-Aided Diagnosis of Prostate Cancer.” *Future Generation Computer Systems* 114:358–67. doi: 10.1016/j.future.2020.08.015.

Luque, Amalia, Alejandro Carrasco, Alejandro Martín, and Ana de las Heras. 2019. “The Impact of Class Imbalance in Classification Performance Metrics Based on the Binary Confusion Matrix.” *Pattern Recognition* 91:216–31. doi: 10.1016/j.patcog.2019.02.023.

Mangkunegara, I. S., and P. Purwono. 2022. “Analysis of DNA Sequence Classification Using SVM Model with Hyperparameter Tuning Grid Search CV.” Pp. 427–32 in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*.

Martinez-Millana, Antonio, Aida Saez-Saez, Roberto Tornero-Costa, Natasha Azzopardi-Muscat, Vicente Traver, and David Novillo-Ortiz. 2022. “Artificial Intelligence and Its Impact on the Domains of Universal Health

Coverage, Health Emergencies and Health Promotion: An Overview of Systematic Reviews.” *International Journal of Medical Informatics* 166:104855. doi: <https://doi.org/10.1016/j.ijmedinf.2022.104855>.

Millar, Peter R., Patrick H. Luckett, Brian A. Gordon, Tammie L. S. Benzinger, Suzanne E. Schindler, Anne M. Fagan, Carlos Cruchaga, Randall J. Bateman, Ricardo Allegri, Mathias Jucker, Jae-Hong Lee, Hiroshi Mori, Stephen P. Salloway, Igor Yakushev, John C. Morris, Beau M. Ances, Sarah Adams, Ricardo Allegri, Aki Araki, Nicolas Barthelemy, Randall Bateman, Jacob Bechara, Tammie Benzinger, Sarah Berman, Courtney Bodge, Susan Brandon, William (Bill) Brooks, Jared Brosch, Jill Buck, Virginia Buckles, Kathleen Carter, Lisa Cash, Charlie Chen, Jasmeer Chhatwal, Patricio Chrem Mendez, Jasmin Chua, Helena Chui, Laura Courtney, Carlos Cruchaga, Gregory S. Day, Chrismary DeLaCruz, Darcy Denner, Anna Diffenbacher, Aylin Dincer, Tamara Donahue, Jane Douglas, Duc Duong, Noelia Egido, Bianca Esposito, Anne Fagan, Marty Farlow, Becca Feldman, Colleen Fitzpatrick, Shaney Flores, Nick Fox, Erin Franklin, Nelly Joseph-Mathurin, Hisako Fujii, Samantha Gardener, Bernardino Ghetti, Alison Goate, Sarah Goldberg, Jill Goldman, Alyssa Gonzalez, Brian Gordon, Susanne Gräber-Sultan, Neill Graff-Radford, Morgan Graham, Julia Gray, Emily Gremminger, Miguel Grilo, Alex Groves, Christian Haass, Lisa Häsler, Jason Hassenstab, Cortaiga Hellm, Elizabeth Herries, Laura Hoechst-Swisher, Anna Hofmann, Anna Hofmann, David Holtzman, Russ Hornbeck, Yakushev Igor, Ryoko Ihara,

Takeshi Ikeuchi, Snezana Ikonovic, Kenji Ishii, Clifford Jack, Gina Jerome, Erik Johnson, Mathias Jucker, Celeste Karch, Stephan Käser, Kensaku Kasuga, Sarah Keefe, William Klunk, Robert Koeppe, Deb Koudelis, Elke Kuder-Buletta, Christoph Laske, Allan Levey, Johannes Levin, Yan Li, Oscar Lopez, Jacob Marsh, Ralph Martins, Neal Scott Mason, Colin Masters, Kwasi Mawuenyega, Austin McCullough, Eric McDade, Arlene Mejia, Estrella Morenas-Rodriguez, John Morris, James Mountz, Cath Mummery, N. eelesh Nadkarni, Akemi Nagamatsu, Katie Neimeyer, Yoshiki Niimi, James Noble, Joanne Norton, Brigitte Nuscher, Ulricke Obermüller, Antoinette O'Connor, Riddhi Patira, Richard Perrin, Lingyan Ping, Oliver Preische, Alan Renton, John Ringman, Stephen Salloway, Peter Schofield, Michio Senda, Nicholas T. Seyfried, Kristine Shady, Hiroyuki Shimada, Wendy Sigurdson, Jennifer Smith, Lori Smith, Beth Snitz, Hamid Sohrabi, Sochenda Stephens, Kevin Taddei, Sarah Thompson, Jonathan Vöglein, Peter Wang, Qing Wang, Elise Weamer, Chengjie Xiong, Jinbin Xu, and Xiong Xu. 2022. "Predicting Brain Age from Functional Connectivity in Symptomatic and Preclinical Alzheimer Disease." *NeuroImage* 256:119228. doi: <https://doi.org/10.1016/j.neuroimage.2022.119228>.

Mirabnahrazam, Ghazal, Da Ma, Cédric Beaulac, Sieun Lee, Karteek Popuri, Hyunwoo Lee, Jiguo Cao, James E. Galvin, Lei Wang, and Mirza Faisal Beg. 2022. "Predicting Time-to-Conversion for Dementia of Alzheimer's Type Using Multi-Modal Deep Survival Analysis."

*Neurobiology of Aging*. doi:  
<https://doi.org/10.1016/j.neurobiolaging.2022.10.005>.

Nguyen, Anh-duc, Seonghwa Choi, Woojae Kim, Sewoong Ahn, Jinwoo Kim, and Sanghoon Lee. 2019. "Distribution Padding in Convolutional Neural Networks." Pp. 4275–79 in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE.

Nguyen, Dong, Hoang Nguyen, Hong Ong, Hoang Le, Huong Ha, Nguyen Thanh Duc, and Hoan Thanh Ngo. 2022. "Ensemble Learning Using Traditional Machine Learning and Deep Neural Network for Diagnosis of Alzheimer's Disease." *IBRO Neuroscience Reports* 13:255–63. doi:  
<https://doi.org/10.1016/j.ibneur.2022.08.010>.

Nyirandayisabye, Ritha, Huixia Li, Qiming Dong, Theogene Hakuzweyezu, and François Nkinahamira. 2022. "Automatic Pavement Damage Predictions Using Various Machine Learning Algorithms: Evaluation and Comparison." *Results in Engineering* 16:100657. doi:  
<https://doi.org/10.1016/j.rineng.2022.100657>.

Oliveira, Roger Alves de, and Math H. J. Bollen. 2023. "Deep Learning for Power Quality." *Electric Power Systems Research* 214:108887. doi: <https://doi.org/10.1016/j.epsr.2022.108887>.

de Oliveira Silva, Felipe, José Vinícius Ferreira, Jéssica Plácido, Paula Sant'Anna, Juliana Araújo, Valeska Marinho, Jerson Laks, and

Andrea Camaz Deslandes. 2019. “Three Months of Multimodal Training Contributes to Mobility and Executive Function in Elderly Individuals with Mild Cognitive Impairment, but Not in Those with Alzheimer’s Disease: A Randomized Controlled Trial.” *Maturitas* 126:28–33. doi: <https://doi.org/10.1016/j.maturitas.2019.04.217>.

Ota, Miho, Yuriko Numata, Ayako Kitabatake, Eriko Tsukada, Tomohiro Kaneta, Takashi Asada, Kohji Meno, Kazuhiko Uchida, Hideaki Suzuki, Tatsumi Korenaga, and Tetsuaki Arai. 2022. “Structural Brain Network Correlations with Amyloid Burden in Elderly Individuals at Risk of Alzheimer’s Disease.” *Psychiatry Research: Neuroimaging* 319:111415. doi: <https://doi.org/10.1016/j.psychresns.2021.111415>.

Peng, Shuai, Hongbo Huang, Weijun Chen, Liang Zhang, and Weiwei Fang. 2020. “More Trainable Inception-ResNet for Face Recognition.” *Neurocomputing* 411:9–19. doi: [10.1016/j.neucom.2020.05.022](https://doi.org/10.1016/j.neucom.2020.05.022).

Poloni, Katia Maria, and Ricardo José Ferrari. 2022. “A Deep Ensemble Hippocampal CNN Model for Brain Age Estimation Applied to Alzheimer’s Diagnosis.” *Expert Systems with Applications* 195:116622. doi: <https://doi.org/10.1016/j.eswa.2022.116622>.

Pustokhin, Denis A., Irina V Pustokhina, Poonam Rani, Vineet Kansal, Mohamed Elhoseny, Gyanendra Prasad Joshi, and K. Shankar. 2021. “Optimal Deep Learning Approaches and

Healthcare Big Data Analytics for Mobile Networks toward 5G.” *Computers and Electrical Engineering* 95:107376. doi: <https://doi.org/10.1016/j.compeleceng.2021.107376>.

Qasim Abbas, S., Lianhua Chi, and Yi-Ping Phoebe Chen. 2023. “Transformed Domain Convolutional Neural Network for Alzheimer’s Disease Diagnosis Using Structural MRI.” *Pattern Recognition* 133:109031. doi: <https://doi.org/10.1016/j.patcog.2022.109031>.

Riad, Rachid, Olivier Teboul, David Grangier, and Neil Zeghidour. 2022. “Learning Strides in Convolutional Neural Networks.” Pp. 1–17 in *International Conference on Learning Representations*.

Sarno, Laura, Daniele Neola, Luigi Carbone, Gabriele Saccone, Annunziata Carlea, Marco Miceli, Giuseppe Gabriele Iorio, Ilenia Mappa, Giuseppe Rizzo, Raffaella Di Girolamo, Francesco D’Antonio, Maurizio Guida, and Giuseppe Maria Maruotti. 2023. “Use of Artificial Intelligence in Obstetrics: Not Quite Ready for Prime Time.” *American Journal of Obstetrics & Gynecology MFM* 5(2):100792. doi: <https://doi.org/10.1016/j.ajogmf.2022.100792>.

Sarwinda, Devvi, Radifa Hilya Paradisa, Alhadi Bustamam, and Pinkie Anggia. 2021. “Deep Learning in Image Classification Using Residual Network (ResNet) Variants for Detection of Colorectal Cancer.” *Procedia Computer Science* 179:423–31. doi:



<https://doi.org/10.1016/j.procs.2021.01.025>.

Shamshirband, Shahab, Mahdis Fathi, Abdollah Dehzangi, Anthony Theodore Chronopoulos, and Hamid Alinejad-Rokny. 2021. "A Review on Deep Learning Approaches in Healthcare Systems: Taxonomies, Challenges, and Open Issues." *Journal of Biomedical Informatics* 113:103627. doi: <https://doi.org/10.1016/j.jbi.2020.103627>.

Singh, Sinam Ajitkumar, Takhellambam Gautam Meitei, and Swanirbhar Majumder. 2020. "Short PCG Classification Based on Deep Learning." Pp. 141–64 in *Deep Learning Techniques for Biomedical and Health Informatics*. Elsevier Inc.

Suha, Sayma Alam, and Tahsina Farah Sanam. 2022. "A Deep Convolutional Neural Network-Based Approach for Detecting Burn Severity from Skin Burn Images." *Machine Learning with Applications* 9(April):100371. doi: [10.1016/j.mlwa.2022.100371](https://doi.org/10.1016/j.mlwa.2022.100371).

Tao, Ping, Pei-En Chen, Jung Tao, Sheau-Ning Yang, Tao-Hsin Tung, and Ching-Wen Chien. 2020. "Correlation between Potentially Inappropriate Medication and Alzheimer's Disease among the Elderly." *Archives of Gerontology and Geriatrics* 87:103842. doi: <https://doi.org/10.1016/j.archger.2019.03.010>.

Tartaglione, Enzo, Carlo Alberto Barbano, Claudio Berzovini, Marco Calandri, and Marco Grangetto. 2020. "Unveiling COVID-19 from CHEST X-Ray with Deep Learning: A Hurdles

Race with Small Data.” *International Journal of Environmental Research and Public Health* 17(18). doi: 10.3390/ijerph17186933.

The Mathworks. 2018. *Introducing Deep Learning with MATLAB*.

Thepade, Sudeep D., Mayuresh Dindorkar, Piyush Chaudhari, and Shalakra Bang. 2022. “Face Presentation Attack Identification Optimization with Adjusting Convolution Blocks in VGG Networks.” *Intelligent Systems with Applications* 16:200107. doi: <https://doi.org/10.1016/j.iswa.2022.200107>.

Thirunavukarasu, Ramkumar, George Priya Doss C, Gnanasambandan R, Mohanraj Gopikrishnan, and Venketesh Palanisamy. 2022. “Towards Computational Solutions for Precision Medicine Based Big Data Healthcare System Using Deep Learning Models: A Review.” *Computers in Biology and Medicine* 149:106020. doi: <https://doi.org/10.1016/j.compbiomed.2022.106020>.

Venkatesh, Shashanka, Suraj Jain, Vishakan Subramanian, and Vishnu Krishnan. n.d. “Alzheimer’s Disease Classification Notebook.” Retrieved (<https://www.kaggle.com/code/vishakansubramanian/alzheimer-s-disease-classification-notebook>).

Vrbančič, Grega, and Vili Podgorelec. 2022. “Efficient Ensemble for Image-Based Identification of Pneumonia Utilizing Deep

CNN and SGD with Warm Restarts.” *Expert Systems with Applications* 187:115834. doi: <https://doi.org/10.1016/j.eswa.2021.115834>.

Wang, Zijie J., Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Polo Chau. 2021. “CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization.” *IEEE Transactions on Visualization and Computer Graphics* 27(2):1396–1406. doi: 10.1109/TVCG.2020.3030418.

Wolf, Tom Nuno, Sebastian Pölsterl, and Christian Wachinger. 2022. “DAFT: A Universal Module to Interweave Tabular Data and 3D Images in CNNs.” *NeuroImage* 260:119505. doi: <https://doi.org/10.1016/j.neuroimage.2022.119505>.

Xiong, Xiaoping, Siding Hu, Di Sun, Shaolei Hao, Hang Li, and Guangyang Lin. 2022. “Detection of False Data Injection Attack in Power Information Physical System Based on SVM–GAB Algorithm.” *Energy Reports* 8:1156–64. doi: 10.1016/j.egyr.2022.02.290.

Zhang, Xiaopeng, Yaru Li, Yan Tao, Yang Wang, Changhua Xu, and Ying Lu. 2021. “A Novel Method Based on Infrared Spectroscopic Inception-Resnet Networks for the Detection of the Major Fish Allergen Parvalbumin.” *Food Chemistry* 337(March 2019):127986. doi: 10.1016/j.foodchem.2020.127986.

Zhu, Yaolin, Huang JiaYI, Yunhong Li, and Wenya

Li. 2022. "Image Identification of Cashmere and Wool Fibers Based on the Improved Xception Network." *Journal of King Saud University - Computer and Information Sciences*. doi: <https://doi.org/10.1016/j.jksuci.2022.09.009>.



# **MONOGRAF: KOMPARASI MODEL TRANSFER LEARNING ALGORITMA CNN PADA PENYAKIT ALZHEIMER**

PURWONO, S.KOM., M.KOM.  
ALFIAN MA ARIF S.T., M.ENG.  
DR. ISWANTO, S.T., M.ENG., IPM  
KHOIRUN NISA, S.KOM., M.KOM.



**Penerbit UHB Press**

ISBN 978-623-88102-3-9 (PDF)



9 786238 810239