

Kelompok 8:

2702223941 - Benedictus Jason

2702224401 - Jocelyn Amelia Winatha

2702210680 - Stanic Dylan

A. Introduction

Proyek ini dilakukan untuk membandingkan dua strategi utama dalam pemanfaatan model Deep Learning pretrained, yaitu feature extraction dan fine-tuning, pada berbagai arsitektur seperti VGG16, ResNet50V2, dan MobileNetV2, dalam konteks klasifikasi citra tumor otak. Dataset yang digunakan adalah Brain Tumor MRI Dataset, yang berisi gambar hasil MRI dengan beberapa kategori tumor. Sebelum pelatihan, dilakukan preprocessing berupa resize agar ukuran gambar seragam sesuai input model, dan augmentasi sederhana seperti rotasi dan flipping untuk meningkatkan ketahanan model terhadap variasi orientasi citra. Nilai piksel juga dinormalisasi agar pelatihan lebih stabil dan cepat konvergen. Langkah-langkah ini memastikan data siap digunakan dalam eksperimen perbandingan performa antara pendekatan feature extraction dan fine-tuning.

B. Workflow Eksperimen○ *Dataset Collection*

Proyek ini menggunakan Brain Tumor MRI Dataset, diperoleh dari Kaggle dengan link <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset>. Dataset ini terisi dengan 7023 gambar citra hasil MRI otak dalam beberapa kategori tumor dengan format file .jpg dan dimensi setiap gambar 512 x 512 pixel. Struktur dataset tersedia dalam susunan berikut; terbagi menjadi 2 folder besar yaitu Training Folder (80% dataset) dan Testing Folder (20% dataset). Setiap folder berisikan 4 folder kecil yang merupakan kumpulan gambar - gambar dengan kategori tumor masing - masing. Ada 4 kelas tumor yang sudah dikategorisasi, **glioma - meningioma - no tumor and pituitary**.

○ *Dataset Preprocessing*

Dataset menggunakan *ImageDataGenerator*, class fungsi yang disediakan dari library tensorflow python untuk melakukan proses preprocessing dan augmentasi sebelum pelatihan. Tahapan yang dilalui dataset secara detail adalah,

- Training Folder dengan *class ImageDataGenerator* dipisahkan dengan rasio 80 : 20 menjadi training dataset dan validation dataset. Setiap gambar pada Training folder juga melalui proses augmentasi gambar, hal ini mencakup pemasangan satu atau lebih filter gambar secara acak (*rotation, zooming, flipping, shifting, brightness, dll.*). Testing Folder dengan *class ImageDataGenerator* juga digunakan 100% sebagai testing dataset, tapi gambar pada folder ini tidak melalui proses augmentasi gambar karena posisinya untuk menilai performa model yang di-*train* secara objektif.
- Semua gambar dataset (*training, validation, testing*) di-*resize* menjadi ukuran 224 × 224 pixel dan diubah ke format RGB 3 channel, keduanya untuk menyesuaikan input

gambar dengan model yang digunakan karena model memiliki karakteristik input yang spesifik untuk berjalan dengan baik.

- Normalisasi informasi pixel gambar melalui *fungsi preprocess_input* sesuai dengan model yang uji untuk menyesuaikan skala piksel dengan standar model *pretrained*. Proses internal fungsi mencakup, *scaling* nilai pixel dari 0 - 255 menjadi 0 - 1 dengan dibagi nilai 255.0, lalu setiap nilai pixel akan dikurangi dengan rata - rata per channel yang biasanya sudah tersedia dari model *pretrained*-nya.

○ Training

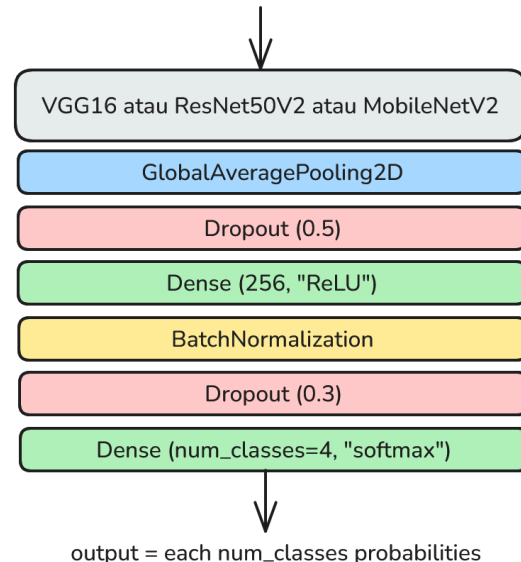
Proses pelatihan dilakukan menggunakan dua pendekatan utama: feature extraction dan fine-tuning, untuk membandingkan performa antara kedua strategi tersebut. Model dasar yang digunakan adalah VGG16, ResNet50V2, dan MobileNetV2 dengan bobot pretrained dari ImageNet dan tanpa fully connected layer (top layer). Dalam pendekatan *feature extraction*, seluruh lapisan konvolusi dibekukan agar hanya head layer yang dilatih, sedangkan pada pendekatan *fine-tuning*, sebagian lapisan akhir dibuka untuk dilatih menyesuaikan fitur *pretrained* terhadap karakteristik dataset MRI. Proses training model menggunakan hyperparameter berikut:

- Batch size: 32
- Epochs: 10 untuk masing-masing tahap
- Optimizer: Adam (learning_rate = 3e-4)
- Loss function: Categorical Crossentropy
- Metrics: Accuracy
- Callbacks: *EarlyStopping* dan *ReduceLROnPlateau* (adaptasi learning rate otomatis)

Head layer tersusun atas:

- Layer GlobalAveragePooling2D untuk mengurangi dimensionalitas fitur input sekaligus mengambil informasi yang penting bagi model
- Dense(256, ReLU) untuk pre-trained model bisa lebih menyesuaikan diri dengan dataset *custom* yang dipakai
- Batch Normalization dan Dropout (0.5 s.d. 0.3) untuk *mencegah overfitting*, meningkatkan generalisasi model
- Dense(num_classes, softmax) untuk output model karena tipe tujuan model yaitu klasifikasi antar 4 kelas

input = image 224 x 224 pixel, format BGR 3 channels



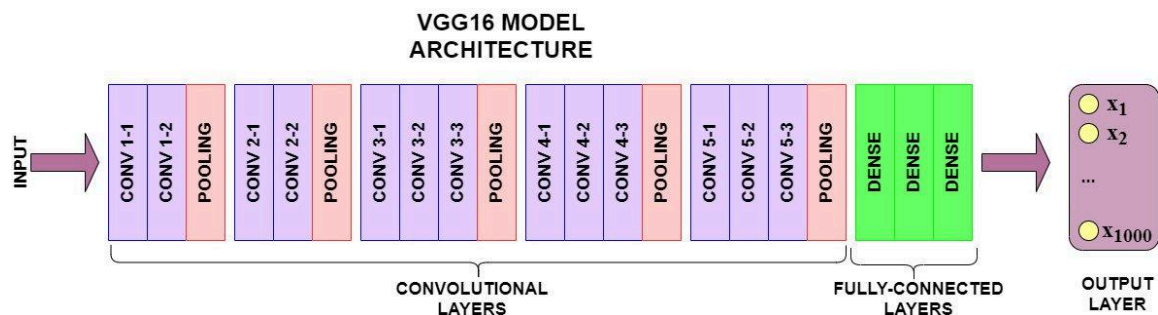
○ Evaluation

Evaluasi model dilakukan menggunakan dataset testing untuk menilai performa akhir. Hasil evaluasi mencakup accuracy, precision, recall, dan F1-score, yang diperoleh melalui classification report. Selain itu, dibuat Confusion Matrix (CM) untuk memvisualisasikan distribusi prediksi antar kelas dan plot kurva training-validation untuk melihat stabilitas pelatihan dan efek fine-tuning terhadap akurasi serta loss. Hasil dari kedua pendekatan,

feature extraction dan fine-tuning dibandingkan untuk mengidentifikasi strategi yang menghasilkan performa terbaik pada klasifikasi citra MRI tumor otak.

C. Analisis Perbandingan Model

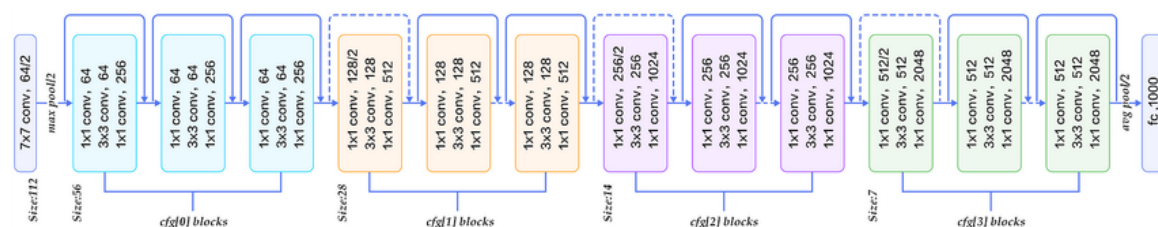
1. VGG16



Gambar diambil dari <https://www.learnatasci.com/tutorials/hands-on-transfer-learning-keras/>

Model VGG16 memiliki struktur konvolusi berlapis-lapis yang sederhana namun sangat dalam. Namun, karena jumlah parameternya yang besar, waktu pelatihan lebih lama dan membutuhkan memori GPU lebih tinggi. Akurasi model baik, tetapi performa cenderung sedikit lebih rendah jika dataset besar dalam segi akurasi dan efisiensi dibandingkan model yang lebih modern seperti ResNet50V2, MobileNetV2, dan lainnya. Hal ini dikarenakan tipe model VGG16 yang *deep network* dan tidak ada usaha mitigasi akan masalah *vanishing gradient*.

2. ResNet50V2



Gambar diambil dari https://www.researchgate.net/figure/Diagram-of-ResNet50V2-architecture-11_fig5_373570488

ResNet50V2 menggunakan konsep residual learning yang memungkinkan jaringan sangat dalam tanpa mengalami *vanishing gradient*. Model cenderung lebih stabil selama proses pelatihan karena adanya skip connections yang membantu propagasi gradien dengan lebih efisien. Gambar di atas bukan bentuk model sebenarnya, namun visualisasi konsep model dimana setiap *residual block* memiliki *skip connection* yang membuat perhitungan output selalu mempertimbangkan input awal *residual block*, dengan itu mengatasi masalah *vanishing gradient*.

3. MobileNetV2

MobileNetV2 dirancang untuk efisiensi dan kecepatan. Arsitektur ini menggunakan *depthwise separable convolution* yang membuat jumlah parameter jauh lebih sedikit dibanding VGG16 dan ResNet50V2. Hasil pelatihan menunjukkan bahwa akurasi model sedikit lebih rendah dari ResNet50V2, tetapi jauh lebih cepat dalam pelatihan dan

prediksi, menjadikannya pilihan ideal untuk deployment pada perangkat dengan sumber daya terbatas seperti smartphone atau edge device.

4. Kesimpulan Perbandingan

Model	Feature Extraction			Fine Tuning		
	Training	Validation	Testing	Training	Validation	Testing
VGG16	0.8090	0.7406	0.8276	0.9798	0.9071	0.9680
	~400s , ~59MB			~430s , ~114MB		
ResNet50V2	0.8078	0.7616	0.8505	0.9809	0.9018	0.9108
	~280s , ~98MB			~290s , ~109MB		
MobileNetV2	0.9298	0.8422	0.8658	0.9622	0.8940	0.9069
	~85s , ~59MB			~90s , ~74MB		

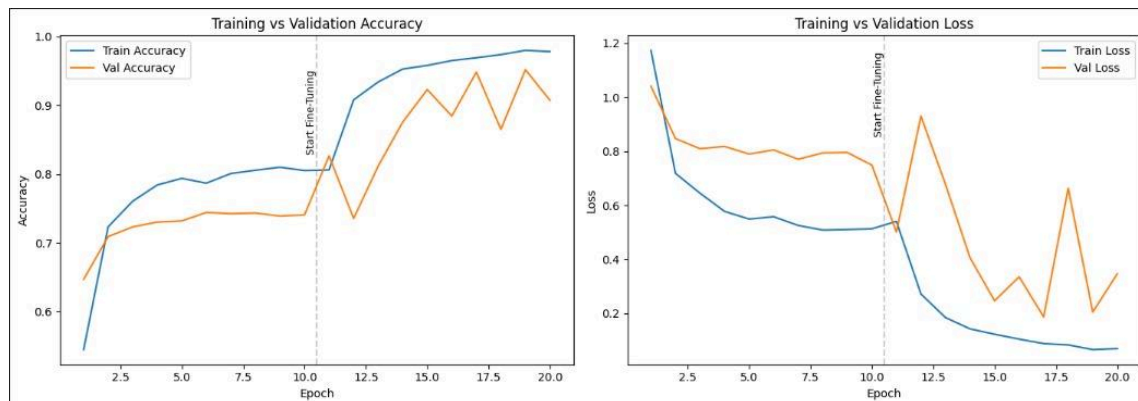
Berdasarkan karakteristik arsitekturnya, ResNet50V2 memiliki kemampuan mencapai akurasi tinggi karena penggunaan residual connections yang membantu aliran gradien saat pelatihan jaringan dalam. MobileNetV2 dirancang dengan arsitektur yang ringan melalui depthwise separable convolution, menjadikannya model dengan pelatihan tercepat dan efisien secara komputasi. Sementara itu, VGG16 memiliki jumlah parameter terbanyak karena strukturnya yang dalam dan padat, sehingga menjadi model terberat di antara ketiganya.

Dalam segi performa tanpa mempertimbangkan hal lain, model VGG16 mencapai akurasi tertinggi dibandingkan model lain, yaitu sebesar 0.9680 dengan catatan *training* model harus menggunakan pendekatan *fine-tuning*. Jika pendekatan yang digunakan adalah *feature-extraction*, maka model MobileNetV2 memiliki performa yang terbaik dengan akurasi sebesar 0.8085. Dalam segi efisiensi waktu dan memori, MobileNetV2 adalah yang terbaik dan teringan, disusul dengan ResNet50V2, lalu yang paling berat dan lambat proses *training*-nya adalah VGG16.

Dalam segi tipe pendekatan yang digunakan, dapat disimpulkan dengan pasti bahwa pendekatan *fine-tuning* meningkatkan performa model dalam melakukan klasifikasi tipe tumor otak. Hal ini disebabkan *pre-trained weight* yang digunakan pada model belum terspesialisasi untuk dataset custom yang dipilih, sehingga apabila digunakan tanpa konfigurasi lanjutan untuk menyesuaikan *weights* pada dataset custom, hasilnya tidak maksimal. Konsep Residual Network juga bisa disimpulkan setara atau meningkatkan akurasi karena dapat dilihat pada hasil *feature-extraction*, kedua MobileNet V2 dan ResNetV2 yang keduanya menggunakan konsep ini, akurasinya lebih tinggi sedikit dibandingkan VGG16.

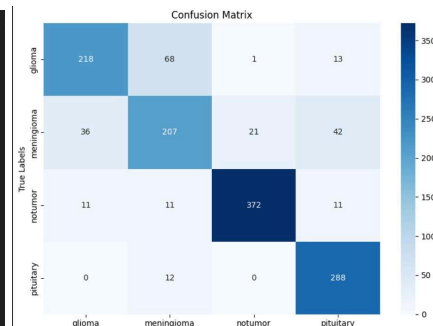
D. Hasil Eksperimen

- VGG16



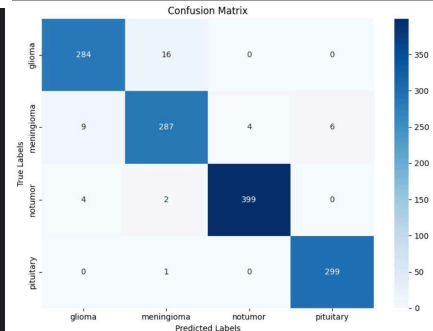
Feature Extraction - Test loss: 0.4922 | Test accuracy: 0.8276
Feature Extraction - Classification Report:

	precision	recall	f1-score	support
glioma	0.82	0.73	0.77	300
meningioma	0.69	0.68	0.69	306
notumor	0.94	0.92	0.93	405
pituitary	0.81	0.96	0.88	300
accuracy			0.83	1311
macro avg	0.82	0.82	0.82	1311
weighted avg	0.83	0.83	0.83	1311

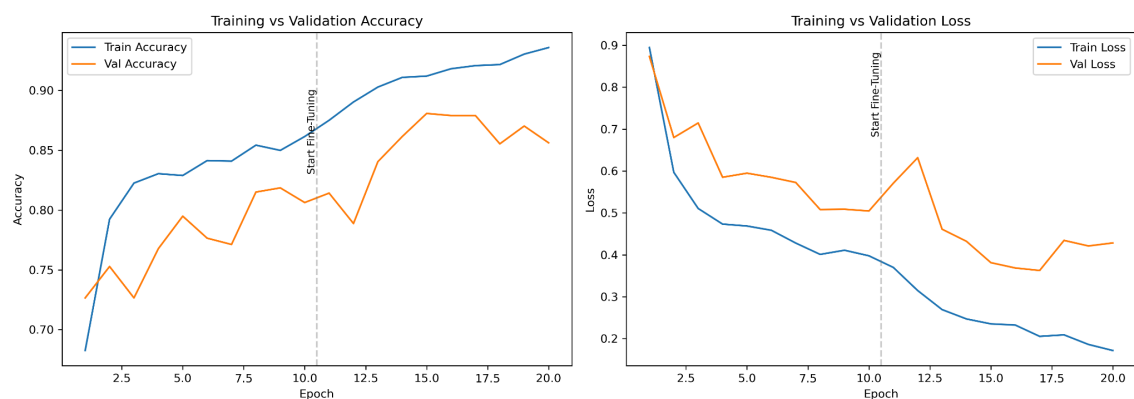


Fine-Tuning - Test loss: 0.1138 | Test accuracy: 0.9680
Fine-Tuning - Classification Report:

	precision	recall	f1-score	support
glioma	0.96	0.95	0.95	300
meningioma	0.94	0.94	0.94	306
notumor	0.99	0.99	0.99	405
pituitary	0.98	1.00	0.99	300
accuracy			0.97	1311
macro avg	0.97	0.97	0.97	1311
weighted avg	0.97	0.97	0.97	1311



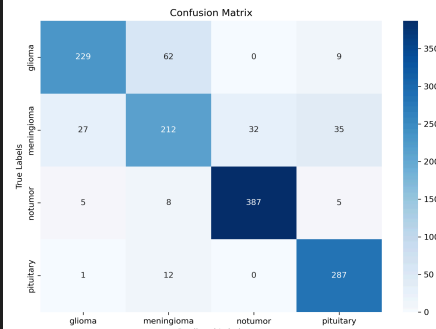
- ResNet50V2



Feature Extraction - Test loss: 0.4154 | Test accuracy: 0.8505

Feature Extraction - Classification Report:

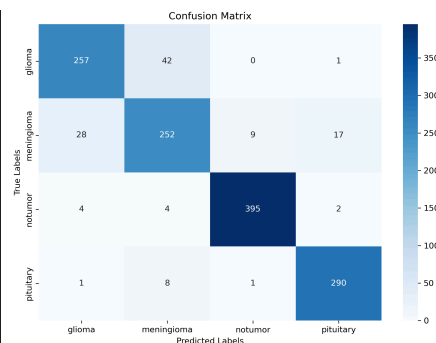
	precision	recall	f1-score	support
glioma	0.87	0.76	0.81	300
meningioma	0.72	0.69	0.71	306
notumor	0.92	0.96	0.94	405
pituitary	0.85	0.96	0.90	300
accuracy			0.85	1311
macro avg	0.84	0.84	0.84	1311
weighted avg	0.85	0.85	0.85	1311



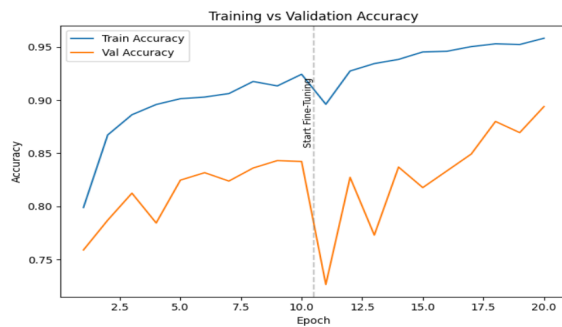
Fine-Tuning - Test loss: 0.2677 | Test accuracy: 0.9108

Fine-Tuning - Classification Report:

	precision	recall	f1-score	support
glioma	0.89	0.86	0.87	300
meningioma	0.82	0.82	0.82	306
notumor	0.98	0.98	0.98	405
pituitary	0.94	0.97	0.95	300
accuracy			0.91	1311
macro avg	0.91	0.91	0.91	1311
weighted avg	0.91	0.91	0.91	1311



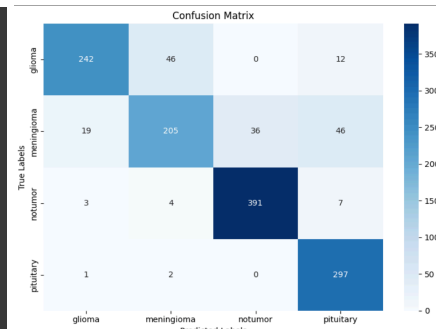
• MobileNetV2



Feature Extraction - Test loss: 0.3899 | Test accuracy: 0.8658

Feature Extraction - Classification Report:

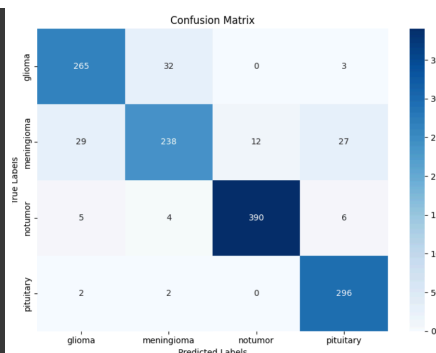
	precision	recall	f1-score	support
glioma	0.91	0.81	0.86	300
meningioma	0.80	0.67	0.73	306
notumor	0.92	0.97	0.94	405
pituitary	0.82	0.99	0.90	300
accuracy			0.87	1311
macro avg	0.86	0.86	0.86	1311
weighted avg	0.87	0.87	0.86	1311



Fine-Tuning - Test loss: 0.2879 | Test accuracy: 0.9069

Fine-Tuning - Classification Report:

	precision	recall	f1-score	support
glioma	0.88	0.88	0.88	300
meningioma	0.86	0.78	0.82	306
notumor	0.97	0.96	0.97	405
pituitary	0.89	0.99	0.94	300
accuracy			0.91	1311
macro avg	0.90	0.90	0.90	1311
weighted avg	0.91	0.91	0.91	1311



Model	Feature Extraction			Fine Tuning		
	Training	Validation	Testing	Training	Validation	Testing
VGG16	0.8090	0.7406	0.8276	0.9798	0.9071	0.9680
	~400s , ~59MB			~430s , ~114MB		
ResNet50V2	0.8078	0.7616	0.8505	0.9809	0.9018	0.9108
	~280s , ~98MB			~290s , ~109MB		
MobileNetV2	0.9298	0.8422	0.8658	0.9622	0.8940	0.9069
	~85s , ~59MB			~90s , ~74MB		

E. Kelebihan & Kekurangan

1. Pendekatan Feature Extraction

Kelebihan:

- **Lebih cepat dan ringan untuk dilatih**, karena seluruh lapisan konvolusi dari model pretrained dibekukan dan hanya lapisan klasifikasi yang dilatih. Hal ini juga terbukti dari memori model saat disimpan dalam format file .keras yang dapat dilihat pada tabel hasil eksperimen.
- **Tidak memerlukan dataset besar**, karena fitur visual dasar sudah diperoleh dari bobot pretrained (misalnya dari ImageNet), pada dasarnya model sudah memiliki kemampuan membaca gambar secara umum.
- **Mengurangi risiko overfitting dan stabilitas pelatihan tinggi**. Hal ini disebabkan jumlah parameter yang dilatih lebih sedikit jika hanya menggunakan pretrained model tanpa *fine-tuning*. Mengingat waktu training yang lebih cepat sedikit, cocok untuk eksperimen awal atau saat sumber daya komputasi terbatas. Hal ini juga terlihat pada grafik akurasi dan loss training, dimana saat fase *feature-extraction* grafik tampak lebih stabil (jarang naik-turun).

Kekurangan:

- **Kurang mampu beradaptasi dengan domain baru**, terutama jika dataset berbeda jauh dari data ImageNet (misalnya citra medis). Hal ini tampak dari perbedaan akurasi antar kedua pendekatan. Pendekatan *fine-tuning* lebih akurat karena lebih spesialisasi pada dataset *custom*.
- **Akurasi maksimum terbatas**, karena fitur yang diekstrak mungkin tidak sepenuhnya relevan untuk task spesifik seperti klasifikasi MRI otak. Alasannya sama dengan ketidakmampuan model untuk beradaptasi dengan dataset baru.
- **Kurang fleksibel untuk eksplorasi fitur tinggi**, karena lapisan konvolusi tidak diperbarui selama pelatihan.

2. Pendekatan Fine-Tuning

Kelebihan:

- **Akurasi lebih tinggi dan fitur lebih relevan dengan domain data**, karena lapisan atas model pretrained disesuaikan ulang dengan karakteristik dataset baru. Dengan demikian, akurasi dan kemampuan generalisasi terhadap citra MRI kemungkinan besar akan meningkat.
- **Memanfaatkan pengetahuan pretrained sekaligus adaptif**, menghasilkan keseimbangan antara efisiensi dan performa. Hal ini disebabkan karena modelnya sudah pretrained, *developer* tidak perlu training model dari awal lagi supaya model bisa baca gambar lagi, cukup mengarahkan model untuk lebih bisa membaca gambar pada domain yang spesifik.

Kekurangan:

- **Waktu pelatihan lebih lama**, karena sebagian besar lapisan ikut dilatih ulang. Hal ini terlihat pada tabel hasil eksperimen yang menunjukkan rata - rata waktu training model dengan pendekatan *fine-tuning* relatif lebih lama sedikit.
- **Berisiko overfitting**, terutama jika dataset kecil dan jumlah parameter yang dibuka terlalu banyak.
- **Membutuhkan GPU dengan kapasitas besar**, karena komputasi bertambah akibat pembaruan bobot di banyak lapisan.
- **Memerlukan tuning hyperparameter yang hati-hati**, seperti learning rate, agar model tidak kehilangan bobot pretrained yang sudah optimal.