

Deteksi Tumor Otak dengan Metode Convolutional Neural Network

Bernadetta Sri Endah Dwi¹, De Rosal Ignatius Moses Setiadi^{*2}

Fakultas Ilmu Komputer, Teknik Informatika
Universitas Dian Nuswantoro
Semarang, Indonesia

e-mail: ¹bernadettadwi14@gmail.com; ²moses@dsn.dinus.ac.id

Diajukan: 7 Juli 2023; Direvisi: 28 November 2023; Diterima: 30 November 2023

Abstrak

Tumor otak merupakan salah satu penyakit mematikan di dunia. Menurut data Global Cancer Observatory, kasus tumor otak di Indonesia pada tahun 2021 mencapai 5.964 kasus serta tingkat kematian berada pada posisi 12 dengan 5298 kasus. Diagnosa cepat dan lebih dini tentu akan mampu menekan tingkat kematian tumor otak. Penelitian ini mengusulkan metode Convolutional Neural Network (CNN) untuk deteksi otak berdasarkan pencitraan medis. Model CNN didesain secara khusus terdiri dari 14 layer. Berdasarkan hasil pengujian model CNN yang dihasilkan memiliki akurasi tinggi yaitu 99%. Selain itu berdasarkan hasil komparasi dengan dataset yang sama, model yang diusulkan 5% lebih unggul dari metode sebelumnya yang menggunakan pre-trained model MobileNetV2.

Kata kunci: Convolutional Neural Network, Deteksi tumor otak, MRI, Machine Learning, Rekognisi citra.

Abstract

Brain tumors are one of the deadliest diseases in the world. According to Global Cancer Observatory data, brain tumor cases in Indonesia reached 5,964 in 2021, with a death rate ranking 12th at 5,298 cases. Swift and early diagnosis can undoubtedly help reduce the mortality rate of brain tumors. This research proposes the use of Convolutional Neural Network (CNN) methods for brain detection based on medical imaging. The CNN model is specifically designed with 14 layers. Based on the testing results, the generated CNN model exhibits high accuracy, reaching 99%. Additionally, in comparison with the same dataset, the proposed model outperforms the previous method using the pre-trained MobileNetV2 model by 5%.

Keywords: Convolutional Neural Network, Brain tumor detection, MRI, Machine Learning, Image recognition.

1. Pendahuluan

Tumor otak menjadi salah satu penyakit mematikan di dunia. Data dari WHO total kasus baru tumor otak di Indonesia mencapai 5.323 kasus serta angka kematian berada di posisi 13 dengan 4.229 kasus[1]. Tumor otak memiliki 3 jenis yaitu tumor glioma, tumor meningioma, dan tumor pituitary. Dokter dapat mengidentifikasi apakah pasien mengidap tumor otak atau tidak dengan menggunakan MRI (*Magnetic Resonance Imaging*) Scan, namun untuk mendapatkan hasil apakah pasien tersebut mengidap tumor otak jenis apa membutuhkan waktu. Oleh karena itu penulis ingin membuat sebuah sistem yang dapat membantu para dokter untuk lebih cepat mengidentifikasi jenis tumor dari pasien tersebut.

Pada beberapa penelitian identifikasi tumor otak memiliki beragam metode atau algoritma yang digunakan. Penelitian pertama yaitu penelitian identifikasi tumor otak menggunakan Fuzzy C-Means, penelitiannya menggunakan datasets berupa gambar yang terdiri dari tumor otak dan normal. Algoritma yang [2] ambil memiliki kelebihan yaitu dapat menghasilkan area segmentasi yang konsisten dan secara akurat mengelompokkan piksel yang termasuk area tumor. Tahap *preprocessing* dapat secara efisien membantu proses segmentasi melalui penghilangan tulang tengkorak pada citra MRI sehingga segmentasi yang dilakukan dapat terfokus ke bagian otak dan tumor saja. Kelemahan yang dimiliki Fuzzy C-Means pada penelitiannya yaitu pada saat segmentasi tumor menunjukkan terdapat dua atau lebih titik tumor yang

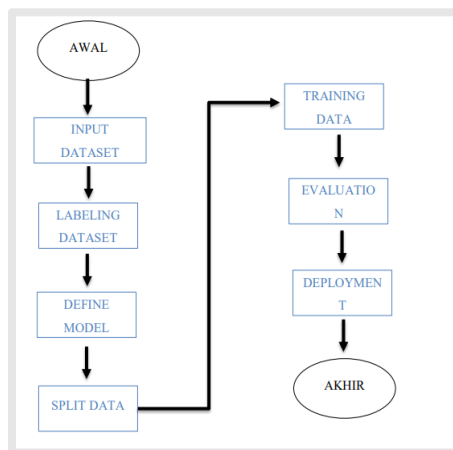
tersegmentasi, karena hasilnya hanya akan ada 1 deteksi area tumor, hal ini disebabkan oleh penggunaan fungsi *morphological filtering* yang hanya memilih area yang terbesar dari hasil segmentasi. Hasil yang didapatkan pada penelitian [2] yaitu tingkat akurasi mencapai 92%. Penelitian [3] juga melakukan identifikasi tumor menggunakan dua Support Vector Machine (SVM), penelitiannya menggunakan datasets berupa gambar yang terdiri dari tumor otak dan normal. SVM memiliki kelebihan yaitu mampu menghasilkan model klasifikasi yang baik meskipun dilatih dengan himpunan data yang relatif sedikit. Kelemahan yang dimiliki SVM yaitu sulit diaplikasikan untuk himpunan dengan jumlah dimensi yang sangat besar. Hasil yang didapatkan pada penelitiannya [3] mencapai 72% untuk tingkat akurasi. Penelitian [4] juga melakukan penelitian identifikasi tumor otak menggunakan Convolutional Neural Network (CNN), datasets yang digunakan berupa gambar yang terdiri dari meningioma tumor, glioma tumor, pituitary tumor, dan normal. Algoritma yang digunakan pada penelitiannya memiliki kelebihan yaitu mampu mengekstraksi fitur kompleks secara otomatis dan efisien untuk klasifikasi citra dengan skala yang besar. Kelemahan yang dimiliki CNN pada penelitiannya yaitu proses *training* atau melatih data model yang cukup lama sehingga penulis perlu memberi *optimizer* dan membuat perancangan model yang tepat. Hasil yang didapatkan pada penelitiannya yaitu tingkat akurasi mencapai 94%. Pada perbandingan beberapa algoritma ini, penulis akan menggunakan algoritma CNN untuk penelitian ini.

Algoritma CNN merupakan jenis jaringan saraf tiruan yang digunakan dalam pengenalan dan pemrosesan gambar yang dirancang khusus untuk memproses data piksel atau data gambar. Secara garis besar CNN tidak jauh beda dengan neural network biasanya [5], [6]. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Convolutional layer juga terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels). Secara umum CNN memiliki kelebihan yaitu dapat secara otomatis mengekstraksi ciri penting dari setiap citra, selain itu metode CNN juga lebih efisien dibandingkan metode neural network lainnya [7]. Kelemahan yang dimiliki oleh CNN yaitu membutuhkan banyak data latih, proses pelatihan (*training*) yang ketiga memakan waktu, dan sering terjadi overfitting. Overfitting merupakan suatu keadaan saat data yang digunakan untuk pelatihan tersebut adalah yang terbaik atau mencapai akurasi 1,000 sehingga apabila dilakukan tes dengan menggunakan data yang berbeda dapat mengurangi akurasi [8]. Namun dengan menggunakan runtime Graphics Pre processing Unit (GPU) dengan kualitas yang baik dapat berguna untuk mempercepat proses *training*. Sedangkan untuk menghindari overfitting dapat menggunakan dropout saat membuat model dan membuat model yang lebih sederhana seperti membangun model dengan dua layer terlebih dahulu dengan maksud mengurangi model yang kompleksitas. Tetapi dengan tingkat kompleksitas yang sangat minimal kemungkinan kinerja CNN masih belum bisa optimal. CNN umumnya setidaknya memiliki lima macam lapisan yang membantu proses berjalannya metode yaitu yang pertama *Input layer*. *Input layer* merupakan tahap pertama dalam metode CNN dimana dalam *Input layer* harus berisi matrix dua dimensi. Lapisan kedua yaitu convolution layer (CL), layer ini digunakan untuk activation map atau mengekstraksi feature [9]. Ketiga yaitu pooling layer (PL), digunakan untuk mengurangi volume spasial citra masukan setelah konvolusi. Keempat yaitu fully connected (FC) layer, digunakan untuk mengklasifikasi gambar antara kategori yang berbeda dengan pelatihan. Terakhir tahap softmax, berguna untuk multi klasifikasi data gambar. Kelima jenis layer ini dapat disusun menjadi sebuah model CNN dimana setiap jenis layer dapat disusun lebih dari satu dengan dapat disusun saling bergantian misalnya pada layer convolution, pooling dan fully connected.

Pre-trained model CNN didesain untuk mempercepat proses pelatihan karena model ini telah dilatih dengan dataset yang besar, sehingga hanya perlu melakukan pelatihan pada bagian akhir layer saja [10]. Tujuannya adalah untuk menghasilkan jaringan syaraf yang ringan melalui pengurangan parameter [11]–[14]. MobileNetV2 adalah model CNN yang didesain ringan untuk komputasi pada processor mobile. Pada penelitian [4] secara khusus menggunakan model pre-trained CNN yaitu MobileNetV2 untuk mendeteksi tumor dan didapatkan akurasi mencapai 94%. MobileNetV2 karena MobileNet-V2 merupakan peningkatan dari versi sebelumnya dan menggunakan teknologi depthwise separable convolution (DSP). Perbedaan MobileNet-V2 dibandingkan dengan versi sebelumnya adalah penambahan fungsi bottleneck dan shortcut connection untuk merangkum lapisan dalam dan berfungsi mengubah konsep dari rendah ke tinggi [15], [16]. Model pre-trained tidak dapat diimplementasikan langsung, proses tuning dan penambahan atau pengurangan layer diperlukan untuk mendapatkan kinerja pre-trained model menjadi lebih optimal. Pada dasarnya pengembangan model CNN dapat disesuaikan dengan kebutuhan, jenis dataset dan tujuannya. Design model CNN secara *custom* baik jumlah, urutan, dan pola layernya harus dilakukan dengan penuh perhitungan dan disesuaikan dengan kebutuhan [17]. Dengan meminimalkan jumlah layer dapat meningkatkan kecepatan proses komputasi [18]. Tetapi perancangan yang kurang baik dan terlalu sederhana dapat menghasilkan performa yang lebih rendah dibandingkan pretrained model CNN. Kontribusi penelitian ini adalah merancang model CNN secara *custom* agar dapat akurat dan menghasilkan performa model CNN yang lebih efisien.

2. Metode Penelitian

Alur penelitian dalam deteksi tumor otak menggunakan algoritma atau metode Convolutional Neural Network terdiri dari beberapa tahapan seperti yang ditunjukkan pada Gambar 1. Tahapan awal penelitian ini dimulai dengan pemrosesan datasets yang termasuk *Input* datasets dan *labelling* dataset. Berikutnya membuat model *custom*. Tahap berikutnya pemisahan datasets, dilanjutkan *training* model lalu evaluasi model dan ditahap akhir deployment model.



Gambar 1. Alur Penelitian.

Penjelasan dari rancangan alur penelitian:

a. *Labelling* Dataset (Pemrosesan Dataset)

Setelah tahap meng*Inputkan* datasets, langkah berikutnya *labelling* datasets. Tahap ini merupakan tahap dimana gambar-gambar pada datasets dikelompokkan sesuai dengan label yang telah ditentukan untuk model aplikasi. Tahap awal dalam *labelling* adalah membuat sebuah fungsi untuk menampung nama-nama dari jenis tumor dan menggunakan OneHotEncoder untuk membuat array yang akan digunakan untuk membantu mengelompokkan datasets lalu folder - folder gambar pada datasets dimasukkan ke dalam nama-nama dari jenis tumor tersebut. Pada tahap ini juga akan dilakukan *resize* gambar menjadi ukuran 128,128,3

b. Define Model

Tahap selanjutnya yaitu membuat model. Tahap ini akan menggunakan library-library keras yaitu seperti conv2d layer, batchnormalization layer, maxpooling 2D layer, dropout, dan lain-lainnya dengan ukuran-ukuran yang telah ditentukan. Untuk mempercepat waktu *training* umumnya akan menggunakan optimizer adam. Menggunakan loss categorical crossentropy karena pada datasets berbentuk folder-folder. Diharapkan model yang telah dibuat dapat menghasilkan tingkat akurasi yang baik.

c. Splitting Data (Pemisahan Data)

Tahap setelah membuat rancangan model yaitu tahap pemisahan data. Data yang telah di*Inputkan* akan dipisahkan menjadi Train Data dan Test Data dengan menggunakan library dari scikit-learn yaitu `train_test_split`. Menggunakan library dari scikit-learn untuk pemisahan data karena Scikit-learn menawarkan berbagai algoritma pembelajaran yang diawasi dan tidak diawasi. Pembelajaran Scikit disusun dengan mempertimbangkan rekayasa perangkat lunak. Desain dasar API bertujuan agar mudah digunakan, namun kuat, dengan tetap mempertahankan fleksibilitas penelitian. Kekokohan ini membuatnya ideal untuk digunakan dalam proyek pembelajaran mesin end-to-end apa pun, mulai dari penelitian hingga produksi.

d. *Training* Model

Tahap berikutnya adalah *training* data. Pada tahap ini *training* akan diatur dengan 100 epoch. Menggunakan 100 epoch hanya perkiraan penulis untuk mencapai target akurasi yang diinginkan namun penulis akan menambah fungsi callback, fungsi callback ini berguna untuk menetapkan akurasi dan validasi akurasi yang diinginkan. Saat akurasi dan validasi akurasi telah mencapai angka yang ditetapkan pada callback maka proses *training* model akan berhenti. Diharapkan dapat menghasilkan tingkat akurasi dan validasi akurasi yang lebih baik dari penelitian sebelumnya.

e. Evaluasi Model

Tahap ini digunakan untuk mengevaluasi tahap sebelumnya, yaitu tahap pelatihan. Tujuannya adalah untuk menyesuaikan model yang telah ditentukan agar sesuai dengan yang diinginkan. Pada tahap ini, model pelatihan yang akan terbentuk adalah algoritma convolutional neural network yang kemudian akan diuji menggunakan confusion matrix untuk mengetahui tingkat akurasi yang diperoleh. Selanjutnya, akan dilakukan pengukuran akurasi, presisi, dan recall untuk mendapatkan total dari tingkat akurasi tersebut. Selain itu, evaluasi waktu juga dilakukan untuk melihat berapa detik yang diperlukan untuk melatih suatu model.

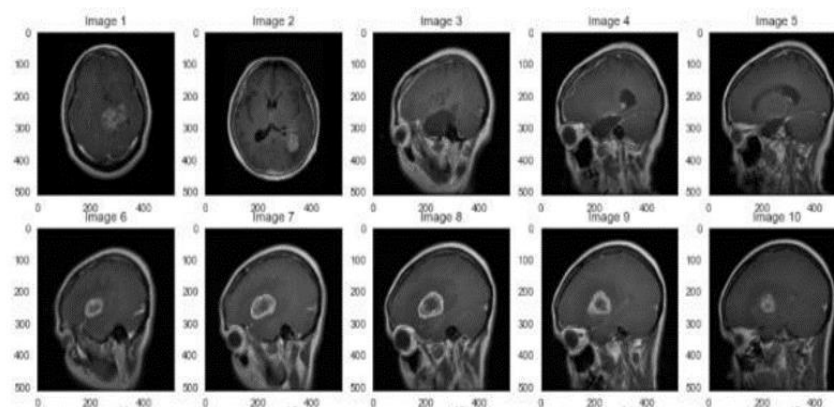
f. Deployment Model

Hasil yang didapat dalam penelitian ini adalah sebuah analisis berbentuk sistem pendukung keputusan yang dapat dipergunakan bagi pihak dokter spesialis tumor otak dalam melakukan deteksi tumor. Penerapan dari hasil algoritma convolutional neural network, diharapkan dapat menjadi referensi dan sumber informasi maupun acuan bagi penelitian selanjutnya yang berkaitan dengan topik yang sejenis.

3. Hasil dan Pembahasan

3.1. Pemrosesan Datasets

Pada tahap ini akan dilakukan pemrosesan datasets dengan menempatkan label data – data dari dataset pada model. Dataset didapat dari website kaggle dengan link <https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri?resource=download>, datasets tersebut memiliki 2870 gambar yang terdiri dari 4 folder yang berisi jenis dari tumor otak yaitu Glioma Tumor (826 gambar) merupakan tumor ganas yang menyerang bagian otak dan sumsum tulang belakang, Meningioma Tumor (822 gambar) merupakan tumor jinak yang menyerang pada selaput yang melindungi otak dan saraf tulang belakang, Pituitary Tumor (827 gambar) merupakan timbulnya sel abnormal yang membentuk massa pada kelenjar pituitari, serta No Tumor atau Normal (395 gambar). Gambar 2 menyajikan sample dataset yang digunakan.



Gambar 2. Dataset Tumor Otak

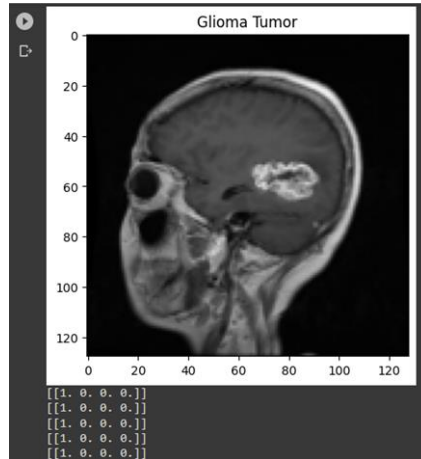
3.1.1. Pembuatan Fungsi

Tahap pertama dalam pemrosesan data yaitu memanggil datasets dari direktori penyimpanan, selanjutnya membuat sebuah fungsi untuk menempatkan label data-data tersebut dengan bantuan OneHotEncoding. OneHotEncoding merupakan salah satu metode encoding yang mempresentasikan data dengan tipe kategorikal sebagai vektor biner bernilai integer. Setelah membuat fungsi tersebut, membuat code untuk menyisipkan data – data gambar sesuai dengan pernyataan kondisi fungsi yang telah dibuat. 4 folder dari datasets tersebut akan dimasukkan ke dalam pernyataan fungsi yang telah dibuat.

3.1.2. Resize Ukuran Citra

Pada tahap ini akan dilakukan pemrosesan datasets dengan cara mengatur ukuran citra gambar dari datasets yang telah dikumpulkan. Tahap ini bertujuan agar saat proses *Input* pengolahan citra dengan algoritma Convolutional Neural Networks (CNN) menjadi sama atau seragam dan mengatasi loss accuracy atau kehilangan tingkat akurasi pada saat proses *training* model. Pada tahap ini, gambar akan di *resize* menjadi ukuran 128 x 128 pixel. Penulis memilih ukuran 128 x 128 pixel karena ukuran tersebut cukup

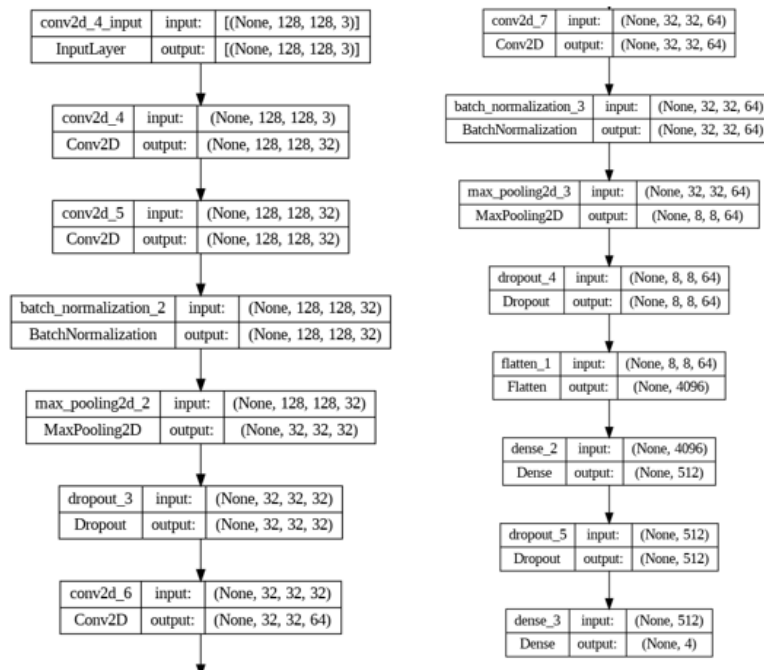
kecil untuk mengurangi daya komputasi yang diperlukan, namun cukup besar untuk mempertahankan detail penting dalam banyak jenis gambar. beberapa model Convolutional Neural Network (CNN) termasuk Xception dirancang untuk gambar berukuran 128x128. Ukuran ini juga sering dipilih karena kompatibilitas dengan operasi pooling dan convolution yang digunakan dalam arsitektur ini. *Resize* citra gambar tersebut berlaku untuk semua ukuran citra gambar pada datasets. Ukuran citra tidak selalu sama maka dari itu dilakukan *resize* agar menjadikan ukuran citra gambar yang sama atau seragam. Sample hasil *resize* disajikan pada Gambar 3.



Gambar 3. Hasil Proses *Resize* Ukuran Dataset

3.2. Pembuatan Model

Setelah tahap pemrosesan data, selanjutnya masuk pada tahap pembuatan model dengan algoritma Convolutional Neural Network (CNN). Pada tahap pembuatan model penulis akan menggunakan model *custom* atau membuat model seutuhnya. Penulis akan menggunakan beberapa layer keras yaitu layer conv2d, batchnormalization layer, maxpooling 2D, dropout, flatten, dan dense. Selain itu digunakan adam sebagai optimizer. Model CNN yang diusulkan memiliki total 14 layer yang disajikan pada Gambar 4.



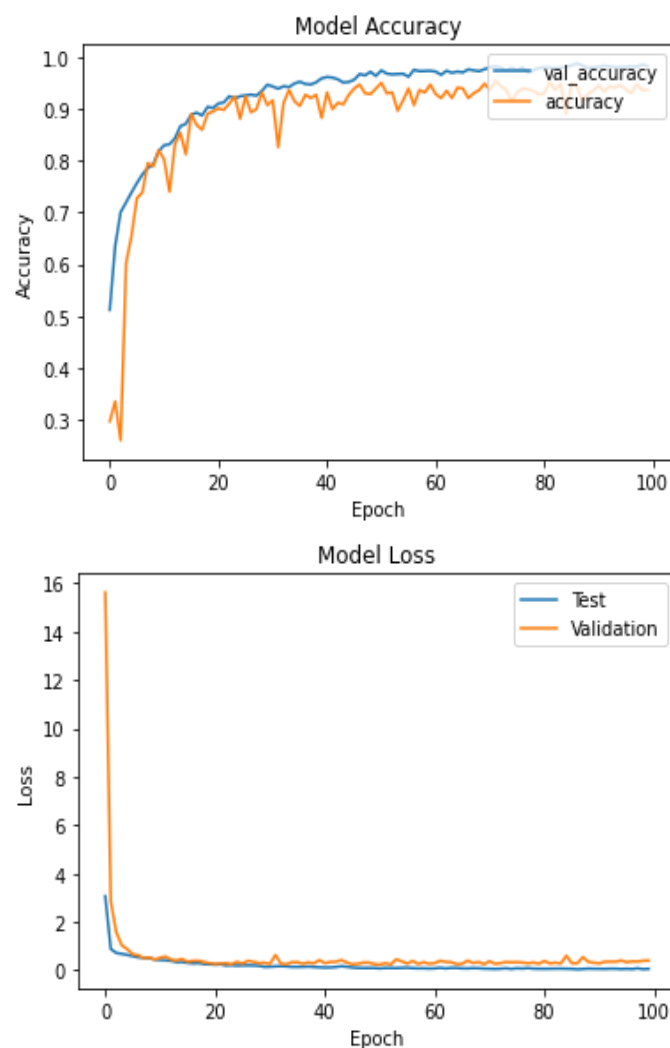
Gambar 4. CNN Model

3.3. Pemisahan Data

Data yang telah diproses sebanyak 2870 data, dibagi menjadi 2 yaitu data latih dan validasi. Pembagian data menggunakan fungsi `train_test_split` dari library `sklearn` yang merupakan sebuah library yang berisi bantuan fungsi untuk memproses data serta membuat model machine learning. Dengan fungsi tersebut data dibagi menjadi 2 bagian yaitu data latih dan validasi. Pembagian dilakukan dengan membagi data latih dengan data validasi. Data dibagi dengan distribusi 80% data latih, 20% data validasi. Hasil dari pembagian tersebut yaitu data latih memiliki total data latih sebesar 2296, dan data validasi sebesar 574.

3.4. Training Model

Proses pelatihan menggunakan GPU dari Google Colab dengan total waktu durasi *training* 8600 detik sebanyak 100 epoch. Dalam pemilihan epoch, membutuhkan waktu eksperimen berulang kali untuk menentukan epoch yang sesuai dan dapat menghasilkan akurasi model yang diinginkan. Proses pelatihan model penelitian membutuhkan 100 epoch untuk mencapai akurasi 99%. Proses pelatihan dilakukan dengan melatih menggunakan data *training* dan menggunakan data testing untuk data validasi dengan tujuan untuk melihat performa model saat proses pelatihan di setiap iterasi. Per epoch membutuhkan waktu sekitar 86 detik untuk menyelesaikan 1 epoch. Perkembangan akurasi serta loss dapat dilihat pada Gambar 5.



Gambar 5. Perkembangan Accuracy dan Loss

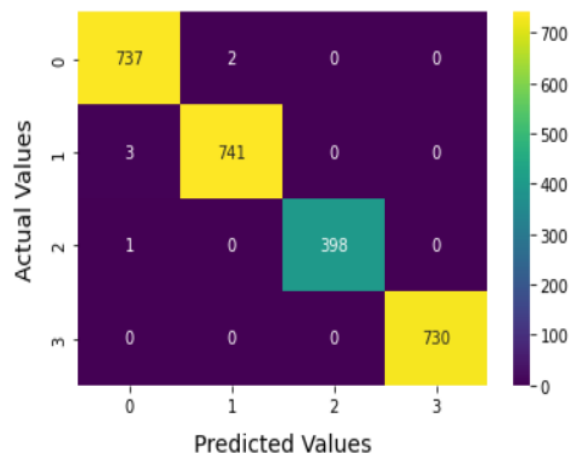
3.5. Evaluasi Model

Evaluasi terhadap model dilakukan dengan melakukan prediksi pada data testing dan melihat hasil confusion matrix, precision, recall, dan F1 score yang didapatkan dari model. Dari hasil proses pelatihan tersebut model mendapatkan akurasi sebesar 98.20% dan loss sebesar 0.06% pada epoch ke 100 serta

menghabiskan 463.231 detik untuk melatih model. Model mengalami peningkatan akurasi setiap iterasinya dan penurunan loss seperti yang diharapkan. Hal ini dapat dilihat pada Gambar 6 yang menunjukkan confusion matrix model dan Tabel 2 yang menunjukkan jumlah precision, recall, F1 score. Angka-angka pada confusion matrix memiliki arti yaitu:

1. True Positives (TP): Ini adalah jumlah kasus ketika kelas sebenarnya dari data point adalah 1 (True) dan prediksi juga 1 (True).
2. True Negatives (TN): Ini adalah jumlah kasus ketika kelas sebenarnya dari data point adalah 0 (False) dan prediksi juga 0 (False).
3. False Positives (FP): Ini adalah jumlah kasus ketika kelas sebenarnya dari data point adalah 0 (False) dan prediksi adalah 1 (True). Secara umum, model memprediksi mereka sebagai kelas positif tetapi mereka sebenarnya bukan.
4. False Negatives (FN): Ini adalah jumlah kasus ketika kelas sebenarnya dari data point adalah 1 (True) dan prediksi adalah 0 (False). Secara umum, model memprediksi mereka sebagai kelas negatif tetapi mereka sebenarnya positif.

Dengan kata lain, angka-angka di matriks kebingungan mencerminkan seberapa baik model melakukan klasifikasi. Angka-angka ini digunakan untuk menghitung matrik evaluasi lebih lanjut seperti Akurasi, Presisi, Recall, dan F1 Score. Selanjutnya kami melakukan komparasi dengan penelitian [4] yang disajikan pada Tabel 3.



Gambar 6. Confusion Matrix dari Model

Tabel 1. Distribusi Jenis Tumor

Glioma Tumor	A
Meningioma Tumor	B
No Tumor	C
Pituitary Tumor	D

Tabel 2. Perhitungan Hasil Klasifikasi

	Precision	Recall	F1-Score
A	$737/741 = 0.99$	$737/739 = 0.99$	$2*737/(741+739) = 0.99$
B	$741/743 = 0.99$	$741/744 = 0.98$	$2*741/(743+744) = 0.99$

C	$398/398 = 1.00$	$398/399 = 0.99$	$2*398/(398+399) = 0.99$
D	$730/730 = 1.00$	$730/730 = 1.00$	$2*730/(730+730) = 1.00$
Accuracy = $(737+741+398+730)/2612 = 0.99$			

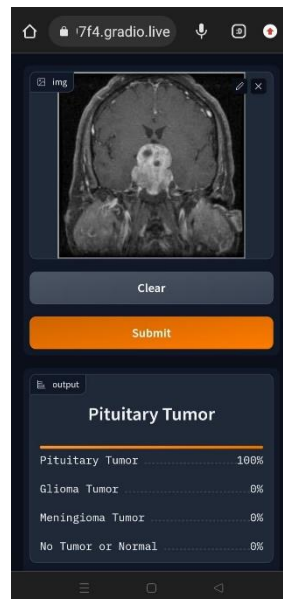
Tabel 3. Komparasi Performa Akurasi, Precision, Recall dan F1-score

	Penelitian Ini				Penelitian [4]			
	Precision	Recall	F1-Score	Akurasi	Precision	Recall	F1-Score	Akurasi
A	0.99	0.99	0.99	737	0.99	0.86	0.92	709
B	0.99	0.98	0.99	741	0.85	0.96	0.90	792
C	1.00	0.99	0.99	398	0.99	0.98	0.99	387
D	1.00	1.00	1.00	730	0.96	0.97	0.97	804
	Support			2612	Support			2870
	Akurasi Akhir			0.99	Akurasi Akhir			0.94

Dalam perbandingan performa antara penelitian ini dan penelitian [4], yang disajikan pada Tabel 3, tampak bahwa model CNN yang diusulkan lebih unggul dalam sebagian besar metrik evaluasi. Dengan presisi yang konsisten lebih tinggi, recall yang lebih baik, dan skor F1 yang lebih tinggi untuk setiap kategori (A, B, C, D). Model CNN yang diusulkan juga memiliki kemampuan yang lebih baik dalam mendeteksi dan mengklasifikasikan tumor otak berdasarkan data medis. Akurasi akhir yang mencapai 0.99 menandakan tingkat kinerja keseluruhan yang sangat tinggi. Di sisi lain, Penelitian [4] menunjukkan performa yang lebih rendah, terutama dalam hal presisi dan recall. Padahal dalam kasus medis, recall lebih penting untuk mendeteksi kasus positif[19]. Hasil ini memberikan implikasi positif terhadap upaya diagnosis dan penanganan tumor otak, dengan model CNN yang diusulkan berpotensi menjadi alat yang lebih efektif dalam mendukung praktisi kesehatan dalam upaya penanganan penyakit mematikan ini.

3.6. Deployment Model

Setelah melakukan evaluasi model, tahap berikutnya yaitu deployment model. Proses deployment pada model penelitian akan diaplikasikan pada web dengan bantuan Gradio, Gradio adalah library python open-source yang digunakan untuk membuat demo model machine learning pada web. Dengan Gradio, dapat dengan cepat membuat user interface yang cantik pada demo model machine learning. Dengan deployment model, orang lain juga dapat turut mencoba demo model kita melalui web atau browser. Sebelum melakukan deployment, perlu menyimpan model dengan format h5 terlebih dahulu. Setelah menyimpan, maka mulai membuat code deployment dengan memasukkan h5 file dan mengunduh library gradio. Hasil deployment model pada web akan memunculkan output berupa gambar yang diinputkan oleh pengguna, jenis tumor otak, serta persentase dari jenis tumor otak tersebut.



Gambar 7. Deployment Model

4. Kesimpulan

Dari penelitian yang telah dilakukan oleh penulis, didapatkan beberapa kesimpulan berdasarkan hasil pengujian. Penelitian ini berhasil mendesign model CNN yang efektif dan efisien untuk recognisi kanker otak. Performanya lebih baik dari metode machine learning model pre-trained yang telah diusulkan sebelumnya yang menggunakan machine learning dan deep learning untuk melakukan deteksi tumor otak. Dengan dataset yang sama, model CNN yang diusulkan dapat lebih unggul 5% daripada model pre-trained MobileNetV2. Dengan akurasi, presisi, recall dan F1-score rata-rata mencapai 99%, model yang memberikan implikasi positif terhadap upaya diagnosis dan penanganan tumor otak berdasarkan *Input* citra MRI pasien. Model ini juga telah berhasil diimplementasikan dengan baik dalam aplikasi mobile-web sehingga dapat dimanfaatkan lebih riil lagi. Kedepan penelitian juga perlu diuji lebih lanjut pada dataset yang lebih besar untuk memvalidasi dan menguji ulang performanya.

Daftar Pustaka

- [1] Global Cancer Observatory, "Global Cancer Observatory in Indonesia," 2021. [Online]. Available: <https://gco.iarc.fr/today/data/factsheets/populations/360-indonesia-fact-sheets.pdf>
- [2] W. S. Sari, C. A. Sari, and R. Setyawan, "Wiener Filter Untuk Optimasi Identifikasi Mri Tumor Otak Berbasis Fuzzy C-Means," 2020, pp. 667–674.
- [3] T. A. Mutiara and Q. N. Azizah, "Klasifikasi Tumor Otak Menggunakan Ekstraksi Fitur HOG dan Support Vector Machine," vol. 4, no. 1, pp. 45–50, 2022, doi: 10.31294/infortech.v4i1.12813.
- [4] W. Hastomo, Sugiyanto, and Sudjiran, "Convolution Neural Network Arsitektur Mobilenet-V2 Untuk Mendeteksi Tumor Otak," in *Ejournal.Jak-Stik.Ac.Id*, 2021, vol. 5, no. 1.
- [5] M. S. Sunarjo and H. Gan, "High-Performance Convolutional Neural Network Model to Identify COVID-19 in Medical Images," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 19–30, Aug. 2023, doi: 10.33633/jcta.v1i1.8936.
- [6] S. B. Imanulloh, A. R. Muslikh, and D. R. I. M. Setiadi, "Plant Diseases Classification based Leaves Image using Convolutional Neural Network," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 1–10, Aug. 2023, doi: 10.33633/jcta.v1i1.8877.
- [7] A. Digdoyo, T. Surawan, A. S. B. Karno, D. R. Irawati, and Y. Effendi, "Deteksi Tumor Otak Dengan CNN Resnet-152," *J. Teknol.*, vol. 9, no. 2, pp. 23–31, May 2022, doi: 10.31479/jtek.v9i2.128.
- [8] S. Deepak and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning," *Comput. Biol. Med.*, vol. 111, p. 103345, 2019, doi: 10.1016/j.combiomed.2019.103345.
- [9] Q. Lina, "Apa itu Convolutional Neural Network?," 2019.
- [10] H. T. Adityawan, O. Farroq, S. Santosa, H. M. M. Islam, M. K. Sarker, and D. R. I. M. Setiadi,

-
- “Butterflies Recognition using Enhanced Transfer Learning and Data Augmentation,” *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 115–128, Nov. 2023, doi: 10.33633/jcta.v1i2.9443.
- [11] F. Rajeena P. P. *et al.*, “A Novel Method for the Classification of Butterfly Species Using Pre-Trained CNN Models,” *Electronics*, vol. 11, no. 13, p. 2016, Jun. 2022, doi: 10.3390/electronics11132016.
- [12] J. J. Chen, J. J. Chen, D. Zhang, Y. Sun, and Y. A. A. Nanehkaran, “Using deep transfer learning for image-based plant disease identification,” *Comput. Electron. Agric.*, vol. 173, no. March, p. 105393, Jun. 2020, doi: 10.1016/j.compag.2020.105393.
- [13] S. K. Behera, A. K. Rath, and P. K. Sethy, “Maturity status classification of papaya fruits based on machine learning and transfer learning approach,” *Inf. Process. Agric.*, May 2020, doi: 10.1016/j.inpa.2020.05.003.
- [14] K. O. Mohammed Aarif and S. Poruran, “OCR-Nets: Variants of Pre-trained CNN for Urdu Handwritten Character Recognition via Transfer Learning,” *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2294–2301, 2020, doi: 10.1016/j.procs.2020.04.248.
- [15] K. Dong, C. Zhou, Y. Ruan, and Y. Li, “MobileNetV2 Model for Image Classification,” *Proc. - 2020 2nd Int. Conf. Inf. Technol. Comput. Appl. ITCA 2020*, pp. 476–480, Dec. 2020, doi: 10.1109/ITCA52113.2020.00106.
- [16] R. Indraswari, R. Rokhana, and W. Herulambang, “Melanoma image classification based on MobileNetV2 network,” *Procedia Comput. Sci.*, vol. 197, pp. 198–207, Jan. 2022, doi: 10.1016/J.PROCS.2021.12.132.
- [17] M. Toğaçar, B. Ergen, and Z. Cömert, “BrainMRNet: Brain tumor detection using magnetic resonance images with a novel convolutional neural network model,” *Med. Hypotheses*, vol. 134, p. 109531, 2020, doi: 10.1016/j.mehy.2019.109531.
- [18] J. Seetha and S. S. Raja, “Brain tumor classification using Convolutional Neural Networks,” *Biomed. Pharmacol. J.*, vol. 11, no. 3, pp. 1457–1461, 2018, doi: 10.13005/bpj/1511.
- [19] F. Mustofa, A. N. Safriandono, and A. R. Muslikh, “Dataset and Feature Analysis for Diabetes Mellitus Classification using Random Forest,” *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 41–48, Sep. 2023, doi: 10.33633/jcta.v1i1.9190.