

Aplikasi Deteksi Tumor Otak Citra MRI Menggunakan Model VGG16

¹Satya Dwi Permana Putra, ²Intan Nur Farida, ³Made Ayu Dusea Widyadara

^{1,2,3} Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ^{*}satyadwipermanaputra04@gmail.com, ¹in.nfarida@gmail.com, ¹madedara@gmail.com

Penulis Korespondens : Satya Dwi Permana Putra

Abstrak— Pendeteksian dini tumor otak sangat penting dalam meningkatkan efektivitas pengobatan dan memperpanjang harapan hidup pasien. Tumor otak dapat bersifat jinak maupun ganas dan keduanya memiliki resiko serius terhadap system saraf pusat. Metode pencitraan MRI (Magnetic resonance Imaging) merupakan standar utama dalam mendeteksi dan menganalisis keberadaan tumor otak. Namun, interpretasi manual citra MRI oleh ahli radiologi dapat memakan waktu dan berpotensi terjadi kesalahan. System ini dirancang untuk mengklasifikasikan citra MRI ke dalam dua kategori yaitu ada tumor dan tidak ada tumor. Proses pelatihan dilakukan menggunakan dataset citra MRI dan menggunakan model VGG16, pada dataset citra MRI yang telah diberi label dengan tahapan pre-processing berupa resizing, normalisasi, thresholding dan edge detection. Model VGG16 dimodifikasi pada bagian fully connected layer agar mampu menyesuaikan klasifikasi biner sesuai kebutuhan. Hasil pelatihan menunjukkan bahwa model mencapai akurasi pengujian sebesar 88.89% Dengan nilai loss 14%. Visualisasi grafik menunjukkan peningkatan akurasi dan penurunan loss yang konsisten pada data pelatihan maupun validasi. Aplikasi ini diharapkan dapat digunakan sebagai alat bantu diagnosis awal dalam mendeteksi tumor otak secara cepat, pbjektif, dan efisien.

Kata Kunci— deteksi MRI, deep learning, VGG16, klasifikasi citra, tumor otak

Abstract— Early detection of brain tumors is crucial in improving treatment effectiveness and increasing patient survival rates. Brain tumors can be either benign or malignant, and both pose serious risk to the central nervous system. Magnetic Resonance Imaging (MRI) is the gold standart for detecting and analyzing the presence of brain tumors. However, manual interpretation of MRI scans by radiologists can be time consuming and prone to error. This system is designed to classify MRI images into two categories: tumor and no tumor. The training process uses a labelled MRI image dataset and employs the VGG16 model, with pre-processing steps including resizing, normalization, thresholding, and edge detection. The VGG16 model is modified in its fully connected layers to support binary classification as required. Training results show that the model achieved a test accuracy of 95.18% with a loss value pf 12.39%. Graphical visualizations demonstrate a consistent increase in accuracy and decrease in loss for both training and validation data. This application is expected to serve as a decision support tool for early diagnosis, enabling fast, objective, and efficient brain tumor detection.

Keywords— MRI detection, deep learning, VGG16, image classification, brain tumor

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Tumor Otak merupakan gangguan Kesehatan yang ditandai dengan pertumbuhan jaringan abnormal di dalam atau di sekitar otak. Keberadaan tumor ini, baik yang bersifat jinak maupun ganas[1] dapat menimbulkan gangguan neurologis hingga resiko kematian. Berdasarkan laporan medis deteksi dan pengobatan dini merupakan faktor penting yang dapat meningkatkan angka kesembuhan pasien.

Saat ini, teknologi MRI menjadi metode pencitraan yang umum digunakan untuk menganalisis struktur otak secara detail. Namun, proses diagnosis dari hasil MRI masih sangat bergantung pada interpretasi manual oleh ahli radiologi yang berpengalaman. Hal ini tidak hanya membutuhkan waktu tetapi juga berisiko menyebabkan variasi hasil akibat faktor kelelahan atau subjektivitas.

Perkembangan teknologi kecerdasan buatan khususnya deep learning[2], membuka peluang besar dalam otomatisasi proses analisis citra medis. Model VGG16 adalah salah satu arsitektur CNN (Convolutional Neural Network)[3] yang telah terbukti efektif dalam tugas klasifikasi citra. Dengan struktur 16 lapisan, model ini mampu mengekstrak fitur visual secara mendalam dan telah banyak digunakan di berbagai bidang termasuk pada bidang medis.

Penelitian ini bertujuan membangun sebuah aplikasi berbasis VGG16 yang mampu mendeteksi keberadaan tumor otak pada citra MRI. System dikembangkan untuk membedakan anatra kondisi adanya tumor ataupun tidak adanya tumor secara otomatis. Aplikasi ini diharapkan dapat menjadi solusi pendukung diagnosis awal kepada pasien yang terkena diagnosis tumor otak

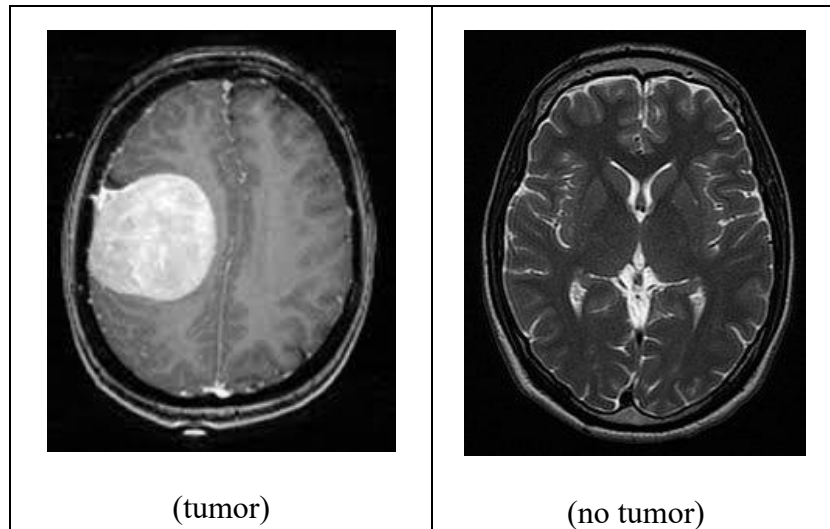
II. METODE

Penelitian menggunakan dataset citra MRI otak yang diklasifikasikan sebagai “yes” tumor dan “no” tanpa tumor. Proses diawali dengan ekstraksi dataset dari file .zip. Kemudian setiap citra melalui rangkaian pra-pemrosesan dan segmentasi. Tahapan ini meliputi konversi ke grayscale, penerapan *gaussian blur* untuk mengurangi *noise*, *Otsu thresholding* [4] untuk binerisasi serta deteksi tepi *canny*[5]. Kontur yang terdeteksi kemudian digunakan untuk menggambar kotak pembatas hijau (*bonding box*) pada area yang berpotensi tumor.

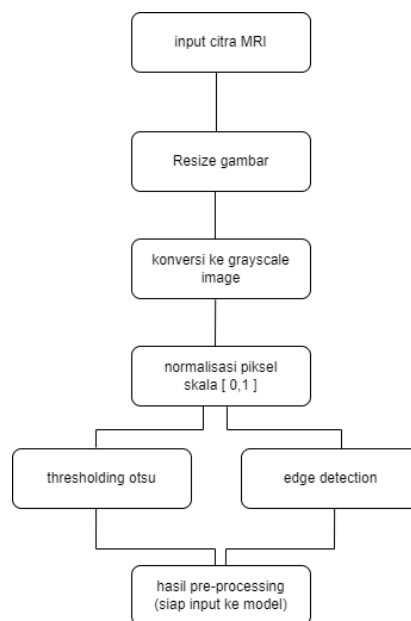
Dataset yang telah tersegmentasi dibagi menjadi data train 70%, data val 20%, dan data tes 10% dengan seed 42 untuk memastikan konsistensi. Data ini kemudian dimuat menggunakan *ImageDataGenerator* dari library keras yang melakukan normalisasi piksel. Model diklasifikasikan dibangun di atas arsitektur VGG16 [6] yang telah dilatih sebelumnya pada *ImageNet*, dengan lapisan dasar dibekukan untuk memanfaatkan pembelajaran fitur yang ada. Ditambahkan lapisan *flatten*, *dense* (256 unit, ReLu), *dropout* (0.5) dan *dense* terakhir (1 unit, *sigmoid*) untuk klasifikasi biner. Model dikompilasi dengan *optimizer* ‘adam’, *loss* ‘binary_crossentropy’ dan metrik ‘accuracy’ pelatihan berlangsung selama 10 *epoch* dengan *modelcheckpoint* menyimpan model terbaik. Evaluasi kinerja model dilakukan pada data uji, menghasilkan akurasi akhir, laporan klasifikasi, dan matrik konfusi untuk analisis detail.

1. Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari citra MRI otak yang diklasifikasikan ke dalam dua kategori, yaitu citra dengan tumor dan citra tanpa tumor dengan jumlah file 253. Dataset diperoleh dari sumber terbuka yaitu kaggle dan untuk pelabelan dataset dilakukan secara manual. Berikut contoh dataset tumor dan tidak ada tumor: (<https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>)



Gambar 1. Dataset



Gambar 2. flowchart pre-processing

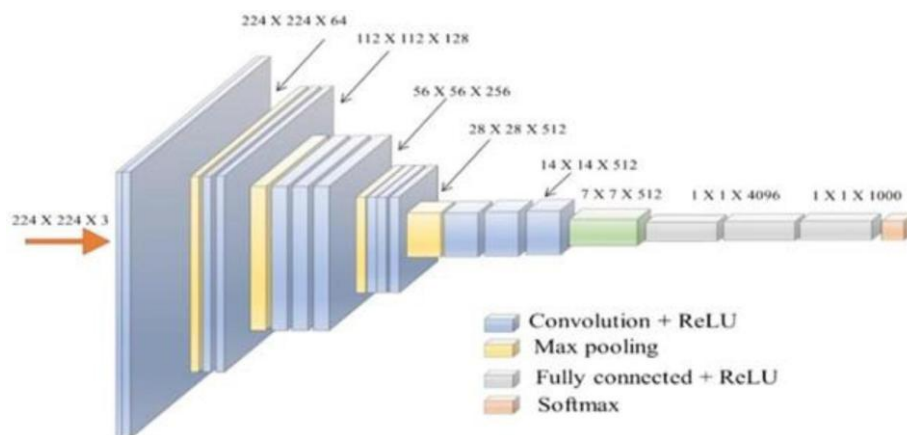
Flowchart pada gambar 2 menjelaskan tahapan *pre-processing* citra MRI sebelum digunakan sebagai input dalam model deteksi tumor otak. Proses dimulai dengan input citra MRI yang kemudian mengalami tahap resizing untuk menyesuaikan ukuran gambar dengan format yang dapat diterima oleh model VGG16 yaitu 224x224 piksel. Setelah itu, citra yang telah di resize dikonversi menjadi *grayscale*[7] *image* agar informasi warna yang tidak relevan dapat dihilangkan dan focus pada tekstur serta intensitas piksel.

Tahap berikut adalah normalisasi piksel ke dalam skala $[0,1]$, yang bertujuan untuk menyamakan rentang nilai piksel dan mempercepat proses pelatihan model. Setelah citra dinormalisasi dilakukan dua teknik pemrosesan lanjutan secara parallel, yaitu thresholding otsu dan edge detection. Thresholding Otsu digunakan untuk membedakan area objek tumor dari latar belakang dengan ambang batas otomatis, sedangkan Edge Detection Canny berguna untuk menyoroti batas-batas objek agar bentuk tumor lebih jelas.

Kedua hasil dari thresholding dan *edge detection* digabungkan sebagai hasil akhir pre-processing, yang siap untuk dimasukkan ke dalam model deep learning VGG16. Tahapan ini bertujuan untuk meningkatkan kualitas fitur citra yang dipelajari model, sehingga dapat memperkuat akurasi dalam klasifikasi citra menjadi dua kategori, yaitu ada tumor dan tidak ada tumor.

2. Arsitektur Model

Vgg16 (*Visual Geometry Group*), yang diperkenalkan oleh Simonyan dan Zisserman pada tahun 2014 menggunakan struktur yang relative sederhana namun efisien dengan lapisan konvolusi yang diatur dalam blok-blok yang diikuti oleh lapisan pooling. Pendekatan ini memungkinkan model untuk menangkap hierarki fitur dari citra, mulai dari fitur dasar seperti tepi dan tekstur[3].



Gambar 3. Arsitektur VGG16

Gambar 3 adalah simulasi dari arsitektur VGG16, pemrosesan data citra yang melalui banyak proses filter sebelum dikelompokkan menjadi beberapa kelompok sesuai klasifikasi[8]. Berikut uraian proses dari VGG16:

- Input $224 \times 224 \times 3$: arsitektur vgg16 menerima input gambar dengan dimensi $224 \times 224 \times 3$, yang menunjukkan bahwa gambar memiliki lebar dan tinggi masing-masing 224 piksel serta 3 saluran warna (RGB)
- Konvolusi + ReLu[9]: lapisan konvolusi pertama terdiri dari 64 filter berukuran 3×3 yang digunakan untuk mendeteksi fitur-fitur dasar seperti tepi, sudut, dan garis-garis dalam gambar. Fungsi aktivasi ReLu (*Rectified Linear Unit*) diterapkan pada keluaran lapisan konvolusi untuk mengaktifkan neuron-neuron dalam jaringan

- c. Max Pooling[9]: setelah lapisan konvolusi terdapat lapisan *max pooling* dengan ukuran kernel 2x2 yang berfungsi untuk meringkas nilai-nilai dalam array gambar
- d. Fully Connected + ReLu [9]: setelah beberapa lapisan konvolusi dan *max pooling* terdapat dua lapisan *fully connected* dengan fungsi aktivasi ReLu. Lapisan ini berfungsi untuk mengekstrak fitur-fitur yang lebih tinggi dari gambar
- e. Softmax[3]: lapisan terakhir adalah *softmax* yang berfungsi untuk menormalisasi vector klasifikasi dan menghasilkan probabilitas kelas untuk klasifikasi gambar

A. Rumus Perhitungan Metrik

Model dievaluasi menggunakan data uji dan metrik evaluasi meliputi akurasi, precision, recall, dan f1-score. Evaluasi dilakukan dengan bantuan library Scikit-learn dan disajikan dalam bentuk classification report serta akurasi dan loss terhadap jumlah epoch pelatihan.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Keterangan:

- a. TP : True Positive
- b. FP : False Positive
- c. FN : False Negative

III. HASIL DAN PEMBAHASAN

Penelitian ini berhasil mengembangkan model deep learning berbasis arsitektur VGG16 untuk melakukan klasifikasi[10] citra MRI otak ke dalam dua kelas, yaitu terdapat tumor dan tidak terdapat tumor. Model dilatih menggunakan dataset citra MRI yang telah melewati tahapan pre-processing guna meningkatkan kualitas citra dan menonjolkan fitur-fitur penting. Proses pelatihan menunjukkan bahwa model mampu mencapai akurasi sebesar 88.89%.

Penerapan metode pre-processing[9] seperti thresholding otsu[11] dan edge detection canny[5] berperan penting dalam membantu model mengenali area tumor dengan lebih jelas serta mengurangi gangguan dari latar belakang citra. Dengan fitur visual yang lebih tajam, model dapat mempelajari karakteristik tumor secara lebih efektif. Evaluasi kinerja dilakukan dengan menggunakan beberapa metrik seperti akurasi, confusion matrix, dan classification report.

Selain pelatihan model, hasil akhir dari penelitian ini diwujudkan dalam bentuk aplikasi web menggunakan framework UI Gradio. Aplikasi ini memungkinkan pengguna mengunggah citra MRI dan memperoleh hasil klasifikasi secara langsung. Penjelasan rincinya sebagai berikut:

1. Hasil pelatihan model

Model dilatih selama 10 *epoch*. Akurasi pelatihan meningkat dari 50.76% pada epoch pertama menjadi 97.13% pada epoch ke sepuluh, sementara loss pelatihan menurun secara signifikan dari 3.5076 menjadi 0.1157. akurasi validasi menunjukkan fluktuasi, mencapai puncaknya 88.00% pada epoch ke-5 dan stabil di sekitar 84.99% pada epoch selanjutnya.

```
self.warn_if_super_not_called()
Epoch 1/10 _____ 0s 17s/step - accuracy: 0.4984 - loss: 3.5734 /usr/local/lib/python3.11/dist-packages/keras/src/trainers/trainer.py:138:
self.warn_if_super_not_called()
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file f
6/6 _____ 140s 24s/step - accuracy: 0.5076 - loss: 3.5076 - val_accuracy: 0.8400 - val_loss: 0.3910
Epoch 2/10 _____ 0s 17s/step - accuracy: 0.7212 - loss: 0.7204 WARNING:absl:You are saving your model as an HDF5 f
6/6 _____ 138s 23s/step - accuracy: 0.7139 - loss: 0.7360 - val_accuracy: 0.8400 - val_loss: 0.2657
Epoch 3/10 _____ 136s 23s/step - accuracy: 0.8138 - loss: 0.4483 - val_accuracy: 0.8400 - val_loss: 0.3565
Epoch 4/10 _____ 132s 23s/step - accuracy: 0.8232 - loss: 0.4802 - val_accuracy: 0.8400 - val_loss: 0.2898
Epoch 5/10 _____ 140s 24s/step - accuracy: 0.8565 - loss: 0.3407 - val_accuracy: 0.8800 - val_loss: 0.2795
Epoch 6/10 _____ 132s 23s/step - accuracy: 0.9004 - loss: 0.1928 - val_accuracy: 0.8400 - val_loss: 0.2858
Epoch 7/10 _____ 143s 25s/step - accuracy: 0.9625 - loss: 0.1631 - val_accuracy: 0.8400 - val_loss: 0.2774
Epoch 8/10 _____ 131s 22s/step - accuracy: 0.9680 - loss: 0.1224 - val_accuracy: 0.8400 - val_loss: 0.2686
Epoch 9/10 _____ 163s 27s/step - accuracy: 0.9674 - loss: 0.1223 - val_accuracy: 0.8400 - val_loss: 0.2665
Epoch 10/10 _____ 143s 25s/step - accuracy: 0.9713 - loss: 0.1157 - val_accuracy: 0.8400 - val_loss: 0.2883
```

Gambar 4. Epoch

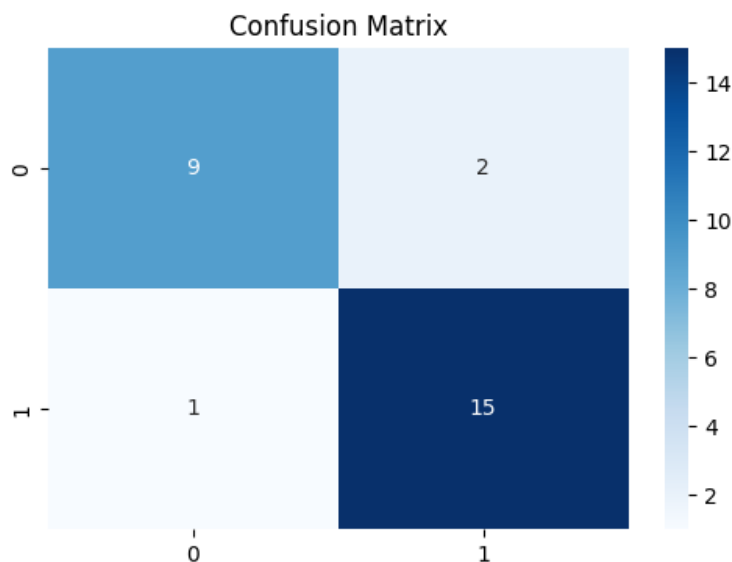
2. Evaluasi model pada data uji

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_data_adapter.py:138:
self.warn_if_super_not_called()
1/1 _____ 30s 30s/step - accuracy: 0.8889 - loss: 0.1471
Akurasi Model di Data Test: 88.89%
```

Gambar 5. Akurasi Model

Pada evaluasi model pada data set data uji menunjukkan akurasi sebesar 88.89%. Data ini diproduksi melalui proses yang dijelaskan di bagian metode, tanpa menyajikan data mentah.

3. Analisis matrik konfusi



Gambar 6. Confussion Matrix

Pada matrik konfusi ini menunjukkan bahwa dari 11 citra tanpa tumor, 9 diklasifikasikan dengan benar, sementara 2 salah diklasifikasikan sebagai tumor (*False Positive*). Dari 16 citra dengan tumor, 15 diklasifikasikan dengan benar, dan hanya 1 salah diklasifikasikan sebagai tanpa tumor (*False Negative*).

4. Laporan klasifikasi

Gambar 7. *Classification Report*

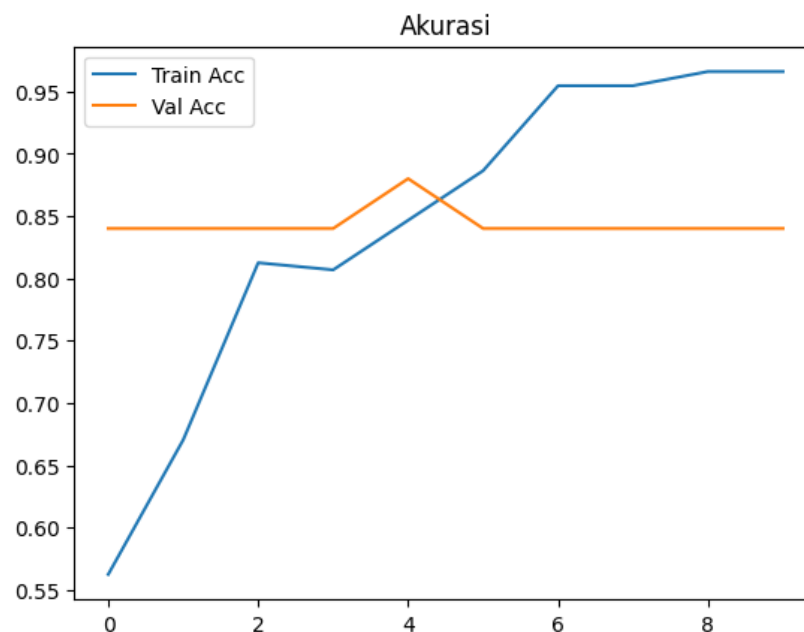
	precision	recall	f1-score	support
no	0.90	0.82	0.86	11
yes	0.88	0.94	0.91	16
accuracy			0.89	27
macro avg	0.89	0.88	0.88	27
weighted avg	0.89	0.89	0.89	27

Gambar 7. Classification Report

Pada laporan klasifikasi ini memberikan metrik evaluasi yang lebih rinci. Untuk kelas 'no' dengan tumor, model menunjukkan *precision* 0.88 dan *recall* 0.94. skor F1 untuk kedua kelas masing-masing adalah 0.86 dan 0.91, menunjukkan keseimbangan yang baik antara *precision* dan *recall*

5. Grafik akurasi dan loss

Gambar 8. Grafik Accuracy dan Loss



Gambar 8. Grafik Accuracy dan Loss

Grafik menunjukkan peningkatan akurasi pelatihan yang konsisten dan penurunan *loss*, mengindikasikan bahwa model belajar secara efektif dari data. Meskipun akurasi validasi

menunjukkan beberapa fluktuasi, tren umumnya stabil, menegaskan kemampuan model untuk menggeneralisasi.

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan sebuah system deteksi tumor otak berbasis citra MRI dengan menggunakan pendekatan deep learning melalui model VGG16. Proses deteksi diawali dengan tahapan pre-processing citra yang mencakup resizing, normalisasi, thresholding otsu, dan edge detection canny untuk memperjelas fitur penting pada citra. Model VGG16 yang telah dimodifikasi untuk klasifikasi biner mampu mengenali keberadaan tumor dengan tingkat akurasi yang tinggi sebesar 88.89% serta nilai loss 14.71%

Implementasi model ke dalam aplikasi web interaktif berbasis UI Gradio turut menambah nilai praktis dari system ini, karena memungkinkan pengguna melakukan deteksi citra secara langsung dengan tampilan visual yang informatif. Secara keseluruhan system ini dapat berfungsi sebagai alat bantu dalam proses diagnosis awal tumor otak sehingga diharapkan dapat mendukung upaya deteksi dini yang cepat efisien dan lebih objektif.

DAFTAR PUSTAKA

- [1] Y. Yueniwati, *Pencitraan pada Tumor Otak*. Universitas Brawijaya Press, 2017.
- [2] S. Khan, N. Islam, Z. Jan, I. Ud Din, and J. J. P. C. Rodrigues, "A novel deep learning based framework for the detection and classification of breast cancer using transfer learning," *Pattern Recognit. Lett.*, vol. 125, pp. 1–6, 2019, doi: <https://doi.org/10.1016/j.patrec.2019.03.022>.
- [3] A. Alshammari, "Construction of VGG16 Convolution Neural Network (VGG16_CNN) Classifier with NestNet-Based Segmentation Paradigm for Brain Metastasis Classification," *Sensors*, vol. 22, no. 20, 2022, doi: 10.3390/s22208076.
- [4] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, 1979, doi: 10.1109/TSMC.1979.4310076.
- [5] E. A. Sekehravani, E. Babulak, and M. Masoodi, "Implementing canny edge detection algorithm for noisy image," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1404–1410, 2020.
- [6] K. R. Krishna, M. Arbaaz, S. N. C. Dhanekula, and Y. M. Vallabhaneni, "Modified VGG16 for accurate brain tumor detection in MRI imagery," *Inform. Autom. Pomiary w Gospod. i Ochr. Środowiska*, vol. 14, no. 3, pp. 71–75, 2024.
- [7] P. D. Hadi, D. A. Widhining K, and F. A. Fiolana, "Identifikasi Jenis Ras Pada Kucing Menggunakan Algoritma Support Vector Machine (SVM)," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 6, no. 1, pp. 77–86, 2024, doi: 10.26905/jasiek.v6i1.10989.
- [8] J. Correa-Morris, Y. Martínez-Díaz, N. Hernández, and H. Méndez-Vázquez, "Novel histograms kernels with structural properties," *Pattern Recognit. Lett.*, vol. 68, pp. 146–152, 2015, doi: <https://doi.org/10.1016/j.patrec.2015.09.005>.
- [9] U. P. Sanjaya, Z. Alawi, A. R. Zayn, and G. P. Dirgantoro, "Optimasi Convolutional

- Neural Network dengan Standard Deviasi untuk Klasifikasi Pneumonia pada Citra X-rays Paru,” *Gener. J.*, vol. 7, no. 3, pp. 40–47, 2023, doi: 10.29407/gj.v7i3.20183.
- [10] H. A. Khan, W. Jue, M. Mushtaq, and M. U. Mushtaq, “Brain tumor classification in MRI image using convolutional neural network,” *Math. Biosci. Eng.*, 2021.
- [11] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.